RANCANG BANGUN APLIKASI IDENTIFIKASI LALU-LINTAS DATA VOICE OVER INTERNET PROTOCOL (VOIP) PADA JARINGAN (STUDI KASUS : SKYPE)

Nama : Anggi Malanda Yoga Putra

NIM : 06.41010.0110

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

SEKOLAH TINGGI

MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER

SURABAYA

2011

RANCANG BANGUN APLIKASI IDENTIFIKASI LALU-LINTAS DATA VOICE OVER INTERNET PROTOCOL (VOIP) PADA JARINGAN (STUDI KASUS : SKYPE)

SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer

Oleh :

Nama : Anggi Malanda Yoga Putra

NIM : 06.41010.010

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

SEKOLAH TINGGI

MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER

SURABAYA

2011

Tidak ada kata menyerah untuk meraih segala sesuatu,

Selalu YAKIN dalam melangkah dan TEGUH berserah diri kepada-Nya.

Karena Sesungguhnya Apa Yang Kita Pikirkan Adalah Apa Yang Kita Dapatkan.

Buku ini penulis persembahkan untuk orang tua, keluarga, kerabat dan pihak terkasih penulis yang tercinta, yang telah memberikan dukungan baik moril maupun materiil, serta kasih dan sayangnya kepada penulis.

TUGAS AKHIR

RANCANG BANGUN APLIKASI IDENTIFIKASI LALU-LINTAS DATA VOICE OVER INTERNET PROTOCOL (VOIP) PADA JARINGAN (STUDI KASUS : SKYPE)

dipersiapkan dan disusun oleh

Anggi Malanda Yoga Putra

NIM : 06.41010.0110

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji
pada : Maret 2011

Susunan Dewan Penguji

Pembimbing

I. Anjik Sukmaaji, S.Kom, M.Eng _____

II. Teguh Sutanto M.Kom, MCP _____

Penguji

I. Harianto, S.Kom, M.Eng _____

II. Rangsang Purnama, M.Kom, MCP _____

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Pantjawati Sudarmaningtyas, S.Kom.,OCA
Wakil Ketua Bidang Akademik

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER SURABAYA

PERNYATAAN

Dengan ini saya menyatakan dengan benar, bahwa Tugas Akhir ini adalah asli karya saya, buka plagiat baik sebagian maupun apalagi keseluruhan. Karya atau pendapat orang lain yang ada dalam Tugas Akhir ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya. Apabila di kemudian hari ditemukan adanya tindakan plagiat pada karya Tugas Akhir ini, maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Surabaya, 07 April 2011

Anggi Malanda Yoga Putra
NIM : 06.41010.0110

ABSTRAK

Aplikasi Skype ini tergolong memiliki keunikan karena dalam sebuah jaringan, Skype dapat memastikan jalur-jalur penghubung untuk tetap terbuka dan secara dinamis berpindah memilih jalur yang terbaik pada saat itu. Dari karakteristik ini dapat dibuat sebuah aplikasi untuk mengidentifikasi lalu-lintas data VoIP pada sebuah jaringan LAN dengan studi kasus aplikasi Skype, yang dilakukan dengan mengamati dan merekam semua paket yang melewati *network interface*, serta mengolah informasi yang terdapat pada *header* dari paket-paket tersebut.

Data paket dikoleksi dengan cara merekam paket yang melintas di jaringan. Kegiatan itu disebut sebagai *sniffing process*. Selanjutnya adalah melakukan *filtering* data berdasarkan hasil dari program *sniffing*. Setelah didapat data yang menunjukan bahwa paket tersebut adalah milik Skype dengan panduan literatur yang ada. Data yang sudah difilter akan disimpan dalam *database* server yang nantinya akan digunakan sebagai laporan kepada pengelola jaringan agar mengetahui *user* Skype yang sedang menggunakan fitur VoIP beserta dengan diagram paket datanya, dimana uji parameter Skype menggunakan TCP *connection* port 80 dan port 443 besaran paket saat *login* adalah 28 Byte. Sedangkan untuk komunikasi VoIP besar paket berkisar antara 80 Byte hingga 640 Byte. Dan port bebas yg digunakan Skype untuk komunikasi VoIP adalah port 1024 hingga port 33033.

Dari hasil pembuatan aplikasi ini diketahui dapat mengidentifikasi lalu-lintas data Skype khususnya VoIP pada sebuah jaringan LAN, sehingga akan memberikan informasi tampilan pengguna Skype yang sedang *online* dan sedang berkomunikasi menggunakan VoIP.

Kata Kunci : Skype, Lalu-Lintas Jaringan, *Monitoring* Jaringan, VoIP

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir berjudul Rancang Bangun Aplikasi Identifikasi Lalu-lintas Data VoIP pada Jaringan (Study Kasus : Skype).

Laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan Program S1 Sistem Informasi di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya (STIKOM).

Penulis mengucapkan terima kasih kepada semua pihak yang telah mendukung hingga terselesaikannya Tugas Akhir ini baik secara moril mapun materiil. Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Orang tua, kerabat, dan keluarga yang selalu mendoakan dan mendukung selama pengerjaan Tugas Akhir.
2. Bapak Anjik Sukmaaji,S.Kom.,M.Eng selaku Dosen Pembimbing I atas segala arahan dan bimbingannya selama pembuatan Tugas Akhir.
3. Bapak Teguh Sutanto M.Kom., MCP selaku Dosen Pembimbing II atas segala saran dan bimbingannya selama pembuatan Tugas Akhir
4. Teman-teman lantai 8 yang berada di ruang OSSC atas kesediaannya membantu penulis dalam uji coba Tugas Akhir ini
5. Pihak terkasih penulis yang telah banyak membantu memberikan semangat baik secara moril maupun materiil, serta atas besarnya dukungan yang diberikan kepada penulis

6. Saudara Reza, yang telah bersedia meminjamkan router modemnya untuk digunakan dalam sidang Tugas Akhir
7. Teman-teman kopma yang bersedia meminjamkan switch berserta kabel LAN untuk digunakan dalam sidang Tugas Akhir
8. Semua sahabat dan teman-teman yang selama ini selalu memberi semangat selama pengerjaan tugas akhir

Serta, kepada semua mahasiswa STIKOM terutama angkatan 2006 dan semua pihak yang telah membantu penulis tetapi tidak dapat disebutkan satu persatu pada kesempatan ini. Semoga Tuhan Yang Maha Esa memberikan limpahan berkah yang setimpal kepada semua pihak yang telah banyak memberikan bantuan, bimbingan ataupun nasihat.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, sehingga sangat diharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan dalam penyelesaian tugas–tugas lain di masa mendatang.

Surabaya, Maret 2011

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Di era globalisasi ini perkembangan teknologi dari waktu ke waktu semakin mengalami banyak peningkatan. Salah satunya ialah teknologi menggunakan jaringan internet untuk mengirimkan data paket suara dari suatu tempat ke tempat yang lain menggunakan perantara protokol IP. Teknologi inilah yang dinamakan teknologi VoIP. VoIP mampu melewatkan trafik suara yang berbentuk paket melalui jaringan IP.

Pada awal perkembangannya, VoIP hanya dapat dipakai antar PC multimedia dengan kualitas rendah. Sesuai dengan perkembangan teknologi, kini VoIP memungkinkan komunikasi antar PC ke telepon dan komunikasi antar telepon dengan kualitas layak sehingga layanan aplikasi berbasis VoIP mulai banyak berdedar bahkan dijual oleh operator-operator telekomunikasi di dunia. Beberapa software aplikasi VoIP antara lain adalah Windows NetMeeting, Skype, dan lain-lain.

Dalam tugas akhir ini penulis menggunakan aplikasi Skype sebagai bahan study kasus. Skype adalah *software* aplikasi komunikasi suara berbasis IP melalui internet antara sesama pengguna Skype. Pada saat menggunakan Skype maka pengguna Skype yang sedang *online* akan mencari pengguna Skype lainnya lalu mulai membangun jaringan untuk menemukan pengguna-pengguna lainnya. Skype memiliki beberapa macam fitur VoIP yang dapat memudahkan pengguna untuk berkomunikasi baik secara data maupun suara. Beberapa di antaranya ialah

Skype dilengkapi dengan *SkypeOut* dan *SkypeIn* yang memungkinkan pengguna Skype untuk berhubungan dengan pengguna telepon konvensional dan telepon genggam. Sehingga pengguna tidak hanya dapat melakukan *chatting* secara *online*, tetapi juga dapat melakukan telepon secara *online*.

Aplikasi Skype ini tergolong memiliki keunikan karena dalam sebuah jaringan, Skype dapat memastikan jalur-jalur penghubung untuk tetap terbuka dan secara dinamis berpindah memilih jalur yang terbaik pada saat itu. Pada saat proses *Login*, paket Skype memiliki keunikan daripada software aplikasi *chatting* yang lainnya, salah satunya yakni karakteristik paket yang terkalkulasi kurang dari 90 *bytes*. Berdasarkan karakteristik-karakteristik inilah dapat diidentifikasi pengguna Skype yang sedang *online* pada jaringan hingga fitur apa yang digunakan oleh pengguna Skype tersebut.

Dari pemaparan di atas, akan dibuat sebuah aplikasi untuk mengidentifikasi lalu-lintas data VoIP pada sebuah jaringan LAN dengan studi kasus aplikasi Skype, yang dilakukan dengan mengamati dan merekam semua paket yang melewati *network interface*, serta mengolah informasi yang terdapat pada *header* dari paket-paket tersebut. Sehingga akan didapat tampilan pengguna Skype yang sedang *online* pada jaringan hingga fitur apa yang digunakan oleh pengguna Skype tersebut. Karena jaringan yang akan dimonitoring adalah berbasis TCP/IP, maka fokus dari tugas akhir ini adalah protokol TCP dan UDP.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan diatas, maka rumusan masalah untuk pembuatan sistem ini, yaitu bagaimana merancang dan

membangun suatu aplikasi yang dapat mengidentifikasi lalu-lintas data Skype khususnya VoIP pada sebuah jaringan LAN.

1.3. Pembatasan Masalah

Batasan masalah dari sistem yang dibahas adalah sebagai berikut:

1. Aplikasi ini dibangun pada jaringan berbasis protokol TCP / IP dengan fokus protokol TCP dan UDP.
2. Aplikasi ini hanya menampilkan *capture* nama komputer dan IP *user* yang sedang *online*.
3. Aplikasi ini tidak membahas secara detail tentang *sniffing*, tetapi hanya membahas tentang pengambilan data pada jaringan LAN.
4. Aplikasi ini hanya mengidentifikasi paket data dan tidak membahas tentang keamanan data dan jaringan.
5. Identifikasi VoIP yang dilakukan hanya pada jaringan LAN.

1.4. Tujuan

Dengan mengacu pada perumusan masalah maka tujuan yang akan dicapai penulis dalam penyusunan Tugas Akhir ini adalah untuk merancang dan membangun suatu aplikasi yang mengidentifikasi lalu-lintas data Skype khususnya VoIP pada sebuah jaringan LAN.

1.5. Manfaat

Adapun manfaat yang dapat penulis ambil dari penyusunan Tugas Akhir ini adalah sebagai berikut:

1. Manfaat Praktis

   Diharapkan dari penyusunan Tugas Akhir ini dapat memberikan bantuan kepada seorang admin jaringan untuk memonitor aktivitas pengguna VoIP khususnya Skype dalam suatu jaringan tertentu.

2. Manfaat Teoritis

   a. Bagi Penulis

      Menjadi sebuah sarana implementasi ilmu yang telah didapat selama masa kuliah.

   b. Bagi Perguruan Tinggi

      Hasil penyusunan Tugas Akhir ini dapat menjadi tambahan perbendaharaan perpustakaan STIKOM Surabaya, sehingga dapat bermanfaat bagi mahasiswa lain yang mencari referensi untuk penyusunanan Tugas Akhir sejenis.

1.6. Sistematika Penulisan

Penulisan Tugas Akhir ini secara sistematika diatur dan disusun dalam lima bab, yaitu:

BAB I : PENDAHULUAN

Berisikan tentang Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, Tujuan, Manfaat serta Sistematika Penulisan laporan Tugas Akhir.

BAB II : LANDASAN TEORI

Berisikan teori-teori yang akan digunakan sebagai landasan dalam desain dan implementasi sistem yaitu Model TCP/IP, Diagram

Alur, *Sniffer* dan Identifikasi lalu-lintas data VoIP pada Skype,

BAB III : PERANCANGAN SISTEM

Berisikan tentang analisis sistem, Blok diagram untuk *sniffing Process*, Blok diagram untuk identifikasi lalu-lintas data VoIP, UML model

BAB IV : IMPLEMENTASI DAN EVALUASI SISTEM

Berisikan tentang cara mengimplementasikan sistem dan pengujian terhadap sistem yang dibuat. Uji coba mencakup *Sniffing Process*, pengkoleksian data dari masing-masing komputer dan identifikasi lalu-lintas data VoIP pada jaringan serta laporan fitur yang digunakan oleh pengguna Skype

BAB V : PENUTUP

Berisikan kesimpulan dari Tugas Akhir, serta saran sehubungan dengan adanya kemungkinan pengembangan sistem pada masa yang akan datang.

BAB II

LANDASAN TEORI

2.1. *Voice over Internet Protocol* (VoIP)

Voice over Internet Protocol (VoIP) dikenal juga dengan sebutan IP Telephony didefinisikan sebagai suatu sistem yang menggunakan jaringan internet untuk mengirimkan data paket suara dari suatu tempat ke tempat yang lain menggunakan perantara protokol IP (Tharom, 2002). Dengan kata lain teknologi ini mampu melewatkan trafik suara yang berbentuk paket melalui jaringan IP. Jaringan IP sendiri adalah merupakan jaringan komunikasi data yang berbasis *packet-switch*.

Voice over Internet Protocol (juga disebut VoIP, IP *Telephony*, Internet *telephony* atau *Digital Phone*) juga adalah teknologi yang memungkinkan percakapan suara jarak jauh melalui media internet. Data suara diubah menjadi kode digital dan dialirkan melalui jaringan yang mengirimkan paket-paket data, dan bukan lewat sirkuit analog telepon biasa.

VoIP merupakan teknologi yang membawa sinyal suara digital dalam bentuk paket data dengan protokol IP. Suara yang masuk diubah dalam bentuk format digital. Telah diketahui bahwa komputer merupakan suatu perangkat digital yang melakukan pengolahan data dalam bentuk *binary digit* (bit). Dengan perkembangan teknologi *Digital Signal Processing* (DSP) telah menghasilkan perangkat yang mampu mengolah sinyal analog (misalnya sinyal audio) sebagai sinyal input dan diolah menjadi sinyal digital dan menghasilkan sinyal keluaran dalam bentuk sinyal analog kembali. Proses ini dilakukan oleh soundcard atau

DSP board. Data dalam format digital akan dikirimkan dalam jaringan internet, akan dibagi dalam paket-paket kecil. Hal ini dapat memudahkan dan mempercepat transportasi. Jadi kalau ada data yang hilang, data tidak perlu dikirim ulang cukup paket-paket yang hilang saja. (Pande K. Sudiarta, 2009)

Panggilan VoIP memiliki dua jenis komunikasi yang menempati jaringan IP antara pemanggil (*calling party*) dan pihak yang dipanggil (*called party*), yaitu aliran informasi pembicaraan dan message-message signaling yang mengontrol hubungan dan arakteristik aliran media. Untuk membawa informasi digunakan *Realtime Transport Protocol* (RTP). Sedangkan untuk pensinyalan terdapat dua standar yang dikeluarkan oleh dua badan dunia, yaitu H.323 yang dikembangkan oleh ITU-T dan *Session itiation Protocol* (SIP) oleh *Internet Engineering Task Force* (IETF).

Tiap paket VoIP terdiri atas dua bagian, yakni *header* dan *payload* (beban). *Header* terdiri atas IP *header*, *Real-time Transport Protocol*, *User Datagram Protocol* (UDP) *header*, dan *link header*.

2.1.1. Skype

Salah satu aplikasi VoIP yang tersedia adalah Skype. Skype adalah *software* aplikasi komunikasi suara berbasis IP melalui internet antara sesama pengguna Skype. Pada saat menggunakan Skype maka pengguna Skype yang sedang online akan mencari pengguna Skype lainnya lalu mulai membangun jaringan untuk menemukan pengguna-pengguna lainnya. Skype memiliki berbagai macam fitur yang dapat memudahkan penggunanya. Skype juga dilengkapi

dengan *SkypeOut* dan *SkypeIn* yang memungkinkan pengguna Skype untuk berhubungan dengan pengguna telepon konvensional dan telepon genggam.

Skype Client (SC) membuka port untuk mendengar TCP dan UDP pada nomor port yang dikonfigurasikan ke dalam kotak dialog koneksinya. SC secara acak memilih nomor port sembari instalasi berlangsung. Sebagai tambahan, SC juga membuka TCP port untuk mendengar pada nomor port 80 (HTTP port), dan nomor port 443 (HTTPS port). Meskipun banyak sekali protokol internet, seperti SIP dan HTTP, namun ada pula TCP yang tidak *default* atau UDP port untuk mendengar (Salman A. Baset, 2004).

Skype menggunakan protokol HTTP untuk berkomunikasi dengan Skype server untuk otentikasi *username/password* dan registrasi dengan Skype *directory server*. Versi modifikasi dari protokol HTTP digunakan untuk berkomunikasi dengan sesama Skype *client*.

Perbedaan utama antara Skype dengan *client* VoIP yang lain adalah bahwa Skype berbasis pada desain *Peer-to-peer* (P2P), daripada model tradisional *client-server*. Hanya *user* terautentifikasi yang tebentuk dari model klasik *Client-Server*, menggunakan mekanisme *public key*. Setelah *user* (dan *client*) telah terautentifikasi, semua pensinyalan yang lebih jauh dibentuk pada jaringan P2P, jadi informasi *user* Skype (seperti *contact list*, status, *preference,* dll.) seluruhnya terdesentralisasi dan terdistribusi di antara P2P node. Peers dalam desain P2P dapat berupa node normal atau supernode. Skype menawarkan beberapa layanan end-*user*: 1) Komunikasi suara, 2). Video Komunikasi, 3) File Transfer, dan 4) Layanan *Chatting*. Panggilan suara dapat juga secara langsung menuju PSTN menggunakan layanan *SkypeIn/SkypeOut*. Selanjutnya didonasikan melalui *End-*

to-End (E2E) dari banyak panggilan termasuk peer Skype dan terminal PSTN. (Dario Bonfiglio, 2009)

Baset mempersembahkan sebuah peninjauan tentang operasi Skype dan jaringan Skype secara keseluruhan. Tidak sama dengan *Client-server* berbasis SIP, Skype menggunakan *overlay* jaringan *peer-to-peer* (P2P), sama dengan *file-sharing* KaZaa. Ada tiga hal yang menjadi komponen utama dalam jaringan Skype :

1. Skype *login server* (LS) adalah salah satu dari sedikit komponen sentral dari jaringan. Setiap *user* terautentifikasi dalam server *login* untuk memperoleh akses ke jaringan.
2. Skype *client* (SC) adalah sebuah *user* pengikut dalam sebuah jaringan. Skype *Client* menyediakan semua *user* secara fungsional untuk mengakses ke jaringan, seperti *login*, inisiasi dan menerima panggilan, *instant messages* dan *file transfer*.
3. Super Node (SN). Banyak Skype *Client* yang dapat juga menjadi sebuah super node, yang menyediakan penambahan secara fungsional ke Super Node dan Skype *Client*. Sebuah supernode menampilkan tugas pengarahan alur seperti contohnya melanjutkan permintaan ke tujuan yang tepat dan menjawab ke *query-query* dari Skype *Client* atau Super node yang lain. Super node juga dapat melanjutkan permintaan *login* dengan kondisi *login server* tidak secara langsung sampai dari sebuah Skype *Client*. Sebagai tambahan, Super Node menyediakan media untuk kemampuan mem*proxy* untuk Skype *Client* yang lain yang hanya memiliki akses internet yang terbatas, memasuki *Network Address Translation* (NAT) atau *firewall* tertentu. Setiap Skype *Client* akan

secara otomatis menjadi Super Node jika menemui beberapa kriteria, seperti *high speed* dan akses internet tanpa batas. Namun umumnya sulit untuk mencegah sebuah supernode dari menjadi sebuah supernode. Untuk *login* ke dalam jaringan, sebuah Skype *Client* harus memanggil setidaknya satu supernode. Beberapa supernode memiliki kode yang sulit untuk masuk ke dalam Skype *Client*. Kecuali untuk beberapa autentifikasi, tempat penyimpana *user list* atau koneksi Skype-to-PSTN, tidak ada server pusat yang lebih jauh dalam jaringan Skype. Baik pensinyalan maupun media trafik yang terenkripsi untuk beberapa jenis koneksi Skype. Hanya ada satu pesan berisikan text kosong (*plain text*) setelah instalasi. (Sven Ehlert, 2006)

2.1.2. Komunitas VoIP

Komunitas pengguna / pengembang VoIP di masyarakat berkembang di tahun 2000. Komunitas awal pengguna / pengembang VoIP adalah “VoIP Merdeka”.”VoIP Merdeka” (VM) dicetuskan oleh Onno W. Purbo. Teknologi yang digunakan oleh "VoIP Merdeka" (VM) adalah H.323 yang merupakan teknologi awal VoIP. Sentral VoIP Merdeka di hosting di Indonesia Internet *Exchange* (IIX) atas dukungan beberapa ISP dan Asossiasi Penyelenggara Jasa Internet Indonesia (APJII). Kode area "VoIP Merdeka" pada saat itu secara aklamasi ditentukan menjadi 6288, tentunya tanpa memperoleh restu dari pemerintah.

Di tahun 2005, Anton Raharja dkk dari ICT Center Jakarta mulai mengembangkan VoIP jenis baru berbasis *Session Initiation Protocol* (SIP).

Teknologi SIP merupakan teknologi pengganti H.323 yang sulit menembus *proxy server*. Di tahun 2006, infrastruktur VoIP SIP dikenal sebagai VoIP Rakyat.

2.2. Sniffer

Sniffer merupakan salah satu teknik yang digunakan untuk mengambil dan menganalisa data di suatu jaringan, sehingga sering digunakan untuk memeriksa dan melakukan tes *network security*. Tersedia banyak *tools* sniffer, sebagai contoh salah satunya yaitu *network* monitor dari Microsoft atau produk *intrusion detection* yang ada saat ini atau berupa sniffer lainnya yang tersedia di *Internet* berupa *freeware*. Produk sniffer itu bekerja dalam *prosmicuous mode* yaitu di mana dia akan mengambil dan menganalisa semua paket yang lewat melalui suatu *network interface* secara *real time* (Harry, 2005).

Untuk komunikasi Skype menggunakan koneksi UDP dan TCP. Sebagai mekanisme *fallback* Skype memiliki kemampuan untuk melakukan koneksi ke port TCP 443 (HTTPS) dan port 80 (HTTP), dengan urutan ini. Untuk mendeteksi jangkauan port yang digunakan telah dilakukan pembatasan beberapa jaringan port untuk memakasa Skype *Client* untuk melakukan kombinasi port yang berbeda untuk *login* ke jaringan Skype.

Untuk trafik UDP, umumnya mengatur port semaunya. Pengguna dapat mengubah nomor port dari Skype *Client* dialog konfigurasi. Jika di*setting*, Skype *Client* yang di jangkau oleh port UDP (dan masuk TCP) trafik. Jika pengguna tidak menentukan port yang akan digunakan, maka Skype *Client* akan memilih semaunya satu port yang akan digunakan selama operasi. Nomor port UDP yang

tetap adalah 33033, untuk menghubungkan *bootstrap* super node statis dari Skype *client* yang terkonfigurasi. (Sven Ehlert, 2006)

Sementara Skype menggunakan kedua koneksi UDP dan TCP, UDP tidak wajib untuk beroperasi. Untuk TCP, Skype *Client* mencoba berkomunikasi dengan host lain dengan menggunkan port diatas 1024. Jika permbatasan *firewall* untuk range ini diberlakukan, maka Skype akan mencoba berkoneki dengan port 443. Jika ini juga gagal, maka Skype akan melakukan koneksi dengan port 80. (Sven Ehlert, 2006)

Idnetifikasi lalu-lintas data VoIP dilakukan dengan cara merekam paket yang melintas di jaringan. Kegiatan itu disebut sebagai *sniffing process*. Ada tiga tipe *sniffing*, yaitu IP-*based sniffing*, MAC-*based sniffing* dan ARP-*based sniffing*. IP-*based sniffing* merupakan yang sering digunakan dalam aktifitas *sniffing*. Dengan mengkondisikan suatu *network card* secara *real time*, dengan mengamati semua *packet* yang dipilih berdasarkan IP *address*. Sedangkan yang kedua adalah mengkondisikan *network card* secara *real time* dan mengamati semua *packet* yang dipilih berdasarkan MAC-*address*.

Yang terakhir adalah ARP-*based sniffing* dan sangat berbeda dengan yang lainya dengan tidak mengkondisikan suatu *network card* pada kondisi *real time*, karena ARP *packet* secara terus menerus akan dikirim ke setiap komputer dalam satu jaringan. Hal ini dikarenakan protokol ARP adalah bersifat stateless atau tidak adanya suatu kehandalan data pada paket-paket yang beredar pada suatu jaringan. (Ragowo, 2008)

Proses identifikasi yang dilakukan penulis adalah dengan cara melakukan filtering data berdasarkan hasil dari program sniffing. Setelah didapat data yang

menunjukan bahwa paket tersebut adalah milik aplikasi Skype dengan panduan literatur yang ada. Data yang sudah difilter akan disimpan dalam database server yang nantinya akan digunakan sebagai laporan kepada pengelola jaringan agar mengetahui jumlah *user* beserta fitur yang digunkan oleh *user* Skype.

Langkah pertama dari algoritma kami adalah mengidentifikasi lalu-lintas yang sudah diketahui, aplikasi non-Skype. Sebuah aplikasi database popular yang digunakan, yang berisi standart TCP dan UDP port komunikasi untuk aplikasi yang dikenal. Port TCP 80 dianggap sebagai pengecualian, karena selain HTTP port tersebut juga digunakan oleh Skype. (Marcel Perenyi, 2007)

2.3. *Transmission Control Protocol / Internet Protocol* (TCP/IP)

TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Berkat prinsip ini tugas masing-masing protokol jadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain sepanjang setiap protokol masih bisa saling mengirim dan menerima data. (Ragowo, 2008)

Model ini terdiri dari empat lapisan, berbeda dengan model *Open System Interconnection* (OSI) yang diprakarsai oleh panitia untuk membuat standarisasi, model TCP/IP ini berasal dari pekerjaan praktis para peneliti yang terlibat dalam membangun arsitektur jaringan. Model TCP/IP tidak memiliki *presentation* dan *Session Layer*. Fungsi dari kedua lapisan ini dapat dilakukan sesuai kebutuhan oleh protokol TCP/IP yang berbeda. Mulai dari atas, lapisan-lapisan dari model TCP/IP yaitu:

1. *Application Layer*, sebagaimana pada model OSI pemakai berinteraksi dengan

aplikasi jaringan pada lapisan ini. Data diterima dari pemakai oleh aplikasi jaringan serta dianggap sebagai komando dan sebagai data yang datang dari luar (peer node). Pada lapisan ini, aplikasi TCP/IP menggunakan model *client*/server.

2. *Transport Layer*, lapisan ini berfungsi mengatur alur data di antara dua node yang saling berhubungan. Protokol yang di-gunakan pada lapisan ini adalah *Transmission Control Protocol* (TCP) untuk penyampaian data.
3. *Network Layer*, data dikirimkan pada lapisan ini. *Internet Protocol* (IP) beroperasi pada lapisan ini untuk menentukan rute yang tidak bergantung kepada medium jaringan. Paket tersebut dilewatkan melalui *internetwork*.
4. *Data Link Layer*, pada lapisan ini data ditransmisikan melalui suatu jaringan tunggal. Data yang berasal dari *Network Layer*, tiba pada jaringan lokal (kadang tidak meninggalkan jaringan sama sekali) dan ditransmisikan ke alamat tujuan.(Suryadi, 1997:10)

2.3.1. Protokol TCP/IP

Pada dasarnya komunikasi data merupakan proses mengirimkan data dari satu komputer ke komputer yang lain. Untuk dapat mengirimkan data, pada komputer harus ditambahkan alat khusus, yang dikenal sebagai *network interface* (*interface* jaringan). Jenis *interface* jaringan ini bermacam-macam, bergantung pada media fisik yang digunakan untuk mentransfer data tersebut.

Dalam proses pengiriman data ini terdapat beberapa masalah yang harus dipecahkan. Pertama, data harus dapat dikirimkan ke komputer yang tepat, sesuai tujuannya. Hal ini akan menjadi rumit jika komputer tujuan transfer data ini tidàk

berada pada jaringan lokal, melainkan di tempat yang jauh. Jika lokasi komputer yang saling berkomunikasi "jauh" (secara jaringan) maka terdapat kemungkinan data rusak atau hilang karenanya, perlu ada mékanisme yang mencegah rusaknya data.

Hal lain yang perlu diperhatikan ialah, pada komputer tujuan transfer data mungkin terdapat lebih dari. satu aplikasi yang menunggu datangnya data. Data yang dikirim harus sampai ke aplikasi yang tepat, pada komputer yang tepat, tanpa kesalahan.

Cara alamiah, untuk menghadapi setiap masalah yang rumit ialah memecahkan masalah tersebut menjadi bagian yang lebih kecil. Dalam memecahkan masalah transfer data, para ahli jaringan komputer pun melakukan hal yang sama. Untuk setiap problem komunikasi data, diciptakan solusi khusus berupa aturanaturan untuk menangani problem tersebut. Untuk menangani semua masalah komunikasi data, keseluruhan aturan ini harus bekerja sama satu dengan lainnya. Sekumpulan aturan untuk mengatur proses pengiriman data ini disebut sebagai *protokol kornunikasi data.* Protokol ini diimplementasikan dalam bentuk program komputer (*software*) yang terdapat pada komputer dan, peralatan komunikasi data lainnya..

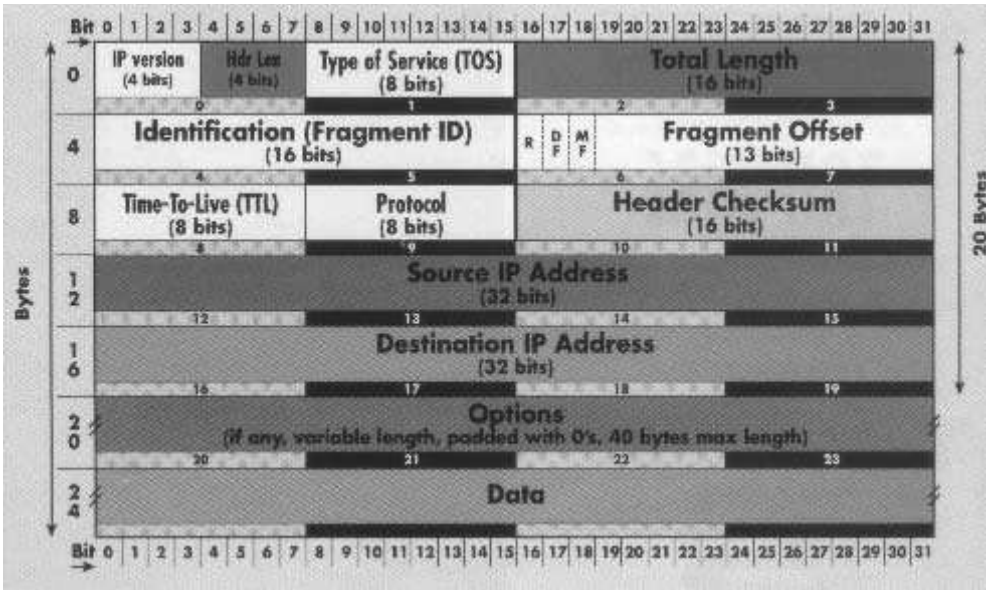
TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungi komunikasi data pada *Wide Area Network* (WAN). TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Berkat prinsip in tugas masing-masing protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain, sepanjang ia masih bisa saling mengirim dan menerima data.

Berkat penggunaan prinsip ini, TCP/IP menjadi protokol komunikasi data yang fleksibel. Protokol TCP/IP dapat diterapkan dengan mudah di setiap jenis komputer dan *interface* jaringan, karena sebagian besar isi kumpulan protokol im tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. Agar TCP/IP dapat berjalan di atas *interface* jaringan tertentu, hanya perlu dilakukan perubahan *path* protokol yang berhubungan dengan *interface* jaringan saja.

2.3.2. *Internet Protocol* (IP) Header

Header Internet Protocol (IP) panjangnya 20 *byte*. Dalam 20 *byte* protokol *header* tersebut adalah beberapa informasi penting yang menjadi jantung proses routing dan pola penyampaian paket dari satu node ke node lain di jaringan Internet. Informasi penting yang digunakan untuk proses routing adalah:

1. *Source IP Address* (32 bit), alamat IP komputer pengirim data
2. *Destination IP Address* (32 bit), alamat IP komputer tujuan
3. *Time-To-Live* (8 bit), jumlah maksimum *router* yang dapat dilewati (maksimum 255 router)
4. Protokol (8 bit), memberitahukan protokol apa yang dibawa di atas IP. Contohnya adalah TCP dan *User Datagram Protocol* (UDP)



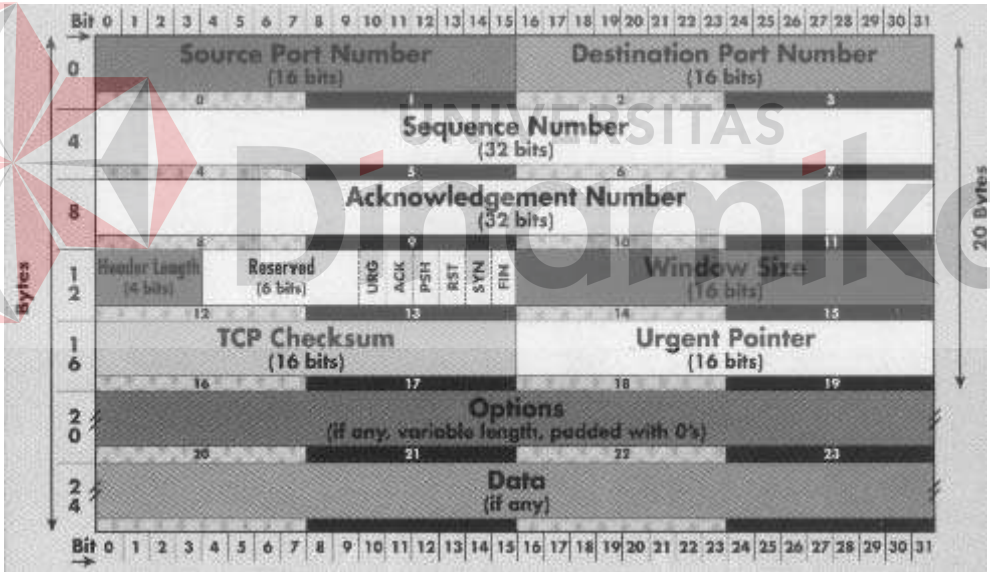
Gambar 2.1 Bagan *Internet Protocol*
(sumber: Ragowo. 2008)

Untuk operasi tidak terlalu normal, misalnya seorang *user* ingin mengirimkan suara telepon melalui jaringan Internet secara *reliable* maka *user* tersebut perlu menset bit pada *Type of Service* (ToS) yang jumlahnya 8 bit. Dalam bahasa sederhana, misalnya bit ToS yang ada dalam IP yang *user* terima adalah 00111100 mempunyai arti bahwa paket tersebut harus di prioritaskan dalam antrian di router, agar memperoleh *delay* seminimal mungkin, dengan *throughput* semaksimal mungkin dan reliabilitas yang tinggi.

Untuk paket yang normal isi *Type of Service* adalah 00000000. IP juga mengontrol apakah data yang *user* kirim dapat di potong menjadi potongan yang lebih kecil, istilahnya difragmentasi. Kontrol apakah paket dapat difragmentasi, dan jika difragmentasi paket tersebut merupakan fragmentasi yang ke berapa akan dikontrol oleh *Flag* di protokol IP. Adapun isi tiga bit pertama dari *Flag* yang kemudian diikuti oleh informasi *offset* Fragmentasi dapat dilihat pada tabel 2.2 (Purbo, 2003:29-31).

2.3.3. *Transmission Control Protocol* (TCP) Header

Transmission Control Protocol (TCP) beroperasi di atas IP. Untuk operasi normal hanya perlu memperhatikan *Source Port* (16 bit), *Destination Port* (16 bit), *Sequence Number* (32 bit) yang merupakan nomor urut paket dan *Acknowledgement Number* (32 bit) merupakan nomor paket yang sudah di acknowledge atau sampai dengan selamat ke tujuan. Perhatikan bahwa *Acknowledgement Number* dan *Sequence Number* merupakan kunci utama dalam menjamin reliabilitas pengiriman data menggunakan protokol TCP. Jika ada paket yang belum diterima atau belum sampai akan dapat dilihat dengan mudah melalui nomor tersebut.



Gambar 2.2 Bagan TCP
(sumber: Ragowo. 2008)

Nomor *port* (16 bit) adalah kode *port* untuk menentukan aplikasi apa yang sedang digunakan. Berbeda dengan IP yang tidak mengenal kondisi sambungan *(stateless)*, pada TCP dikenal *state* (kondisi) sambungan. Yang di maksud kondisi sambungan (state) disini, misalnya klien sedang berusaha menghubungi server, server telah bersedia menerima hubungan komunikasi dengan klien, server dan

klien bertransaksi data, server memutuskan hubungan, klien memutuskan hubungan komunikasi. Kondisi *(state)* tersebut di beberapa kontrol bit seperti pada tabel di bawah ini:

Tabel 2.1 Tabel Kondisi Pada Beberapa Kontrol Bit

Nama	Keterangan
URG (1 bit)	paket yang sifatnya *urgent* / penting
ACK (1 bit)	*acknowledge*, paket diterima dengan baik
PSH (1 bit)	*push*, memaksa
RST (1 bit)	*reset*, mereset hubungan
SYN (1 bit)	sinkronisasi, untukmembuka hubungan awal
FIN (1 bit)	*final*, memutuskan hubungan

Beberapa contoh skenario dalam pemakaian bit kontrol adalah sebagai berikut:

1. Klien mengirimkan SYN ke Server, menandakan klien meminta hubungan dengan server
2. Server menjawab dengan mengirimkan ACK SYN, artinya Server siap menerima hubungan
3. Server dapat juga menjawab dengan mengirimkan PSH RST FIN, artinya server memutuskan hubungan dengan klien dan tidak bersedia menerima hubungan
4. Jika hubungan telah terjalin, biasanya hanya digunakan bit kontrol ACK atau PSH ACK untuk memberikan *acknowledgement* bahwa paket nomor tertentu telah diterima dengan baik

TCP *Checksum* (16 bit) digunakan untuk kode cek apakah paket yang dikirim masih utuh sampai di tujuan atau ada kerusakan. Jika masih utuh maka paket ACK akan dikirimkan oleh penerima ke pengirim. *Window Size* (16 bit) merupakan usaha untuk mengefisienkan penggunaan jaringan dengan cara

mengirimkan beberapa paket sekaligus tanpa menunggu ACK terlebih dulu. Untuk kondisi yang tidak terlalu *reliable* biasanya TCP akan mengirimkan sebuah paket ke tujuan dan menunggu ACK dari tujuan sebelum mengirimkan paket selanjutnya. Konsekuensinya akan ada jeda *(delay)* yang lumayan (bisa beberapa ratus milidetik) antara satu paket dengan paket lainnya. Dengan konsep *sliding window* yang diset maksimum pada *window size* (16 bit), dapat diset *window size* beberapa kali panjang *Maximum Transmission Unit* (MTU). Misalnya MTU 1500 *byte*, dapat diset *window size* 6000 *byte*. Artinya awalah jika pada saat pengiriman data TCP melihat bahwa jaringan cukup *reliable*, artinya tidak ada paket yang hilang/rusak di jalan dan ACK selalu dikirimkan oleh penerima dengan tepat waktu maka TCP akan berusaha mengirimkan beberapa paket sekaligus (2 hingga 4 paket) tergantung panjang *window size* tanpa menunggu ACK dari penerima terlebih dulu.

Tentunya proses menaikkan jumlah paket yang dikirim sekaligus dilakukan secara bertahap, jadi berawal dari satu paket jika ternyata baik maka akan dicoba dengan dua paket, jika sambungan masih baik di coba lagi dengan tiga paket dan seterusnya. Jika terjadi kerusakan pada paket, artinya ACK tidak diperoleh pada waktunya, maka TCP langsung menurunkan lagi paket menjadi satu paket lagi, dan secara bertahap dinaikkan lagi jika sambungan di rasakan *reliable*. Konsep naik turunnya jumlah paket yang dikirimkan secara otomatis ini dinamakan *sliding window*. Tentunya ada beberapa mekanisme lain yang tidak ada pada *header* protokol TCP tapi juga penting untuk menjamin reliabilitas pengiriman data yaitu, konsep TCP *back off* yang menentukan kapan pengiriman ulang sebuah paket yang rusak harus dilakukan. Berbeda dengan *Internet Protocol* (IP) yang

sama sekali tidak menjamin reliabilitas pengiriman data. TCP berusaha secara maksimal agar proses pengiriman data andal dan efisien (Purbo, 2003:29-31).

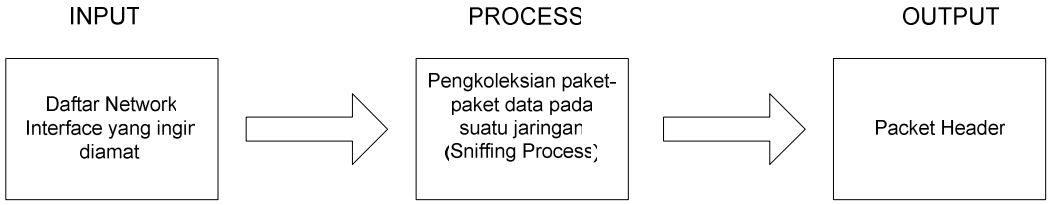
BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1. Analisa Permasalahan

Skype adalah sebuah *software* aplikasi komunikasi suara berbasis IP yang memiliki beberapa macam fitur VoIP sehingga dapat memudahkan pengguna untuk berkomunikasi baik secara data maupun suara. Skype bukan sekedar aplikasi *chatting* biasa, namun aplikasi ini juga memungkinkan *user* untuk berkomunikasi suara dengan sesama *user* Skype layaknya berkomunikasi menggunakan telepon konvensional atau telepon genggam. Sehingga *user* tidak hanya dapat melakukan *chatting* secara *online*, tetapi juga dapat melakukan telepon secara *online* secara gratis.

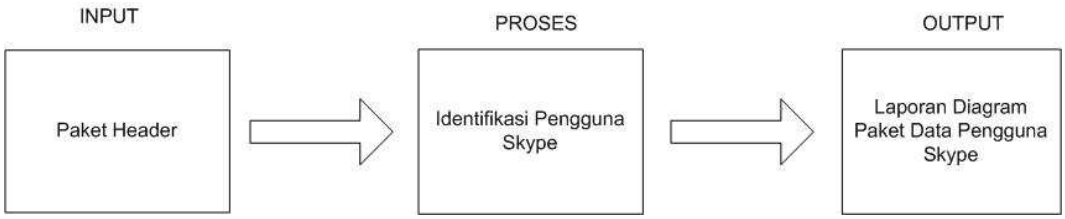
Permasalahan yang dihadapi adalah bagaimana melakukan proses identifikasi VoIP pada aplikasi Skype dalam sebuah jaringan LAN. Untuk langkah pertama, data paket dikoleksi dengan cara merekam paket yang melintas di jaringan. Kegiatan itu disebut sebagai *sniffing process*. Pengkoleksian data tersebut dilakukan oleh *client* atau komputer yang diinstal dengan aplikasi ini, seperti yang terlihat pada gambar di bawah ini.



Gambar 3.1. Blok Diagram untuk *Sniffing Process*

Proses identifikasi yang dilakukan penulis adalah dengan cara melakukan *filtering* data berdasarkan hasil dari program *sniffing*. Setelah didapat data yang

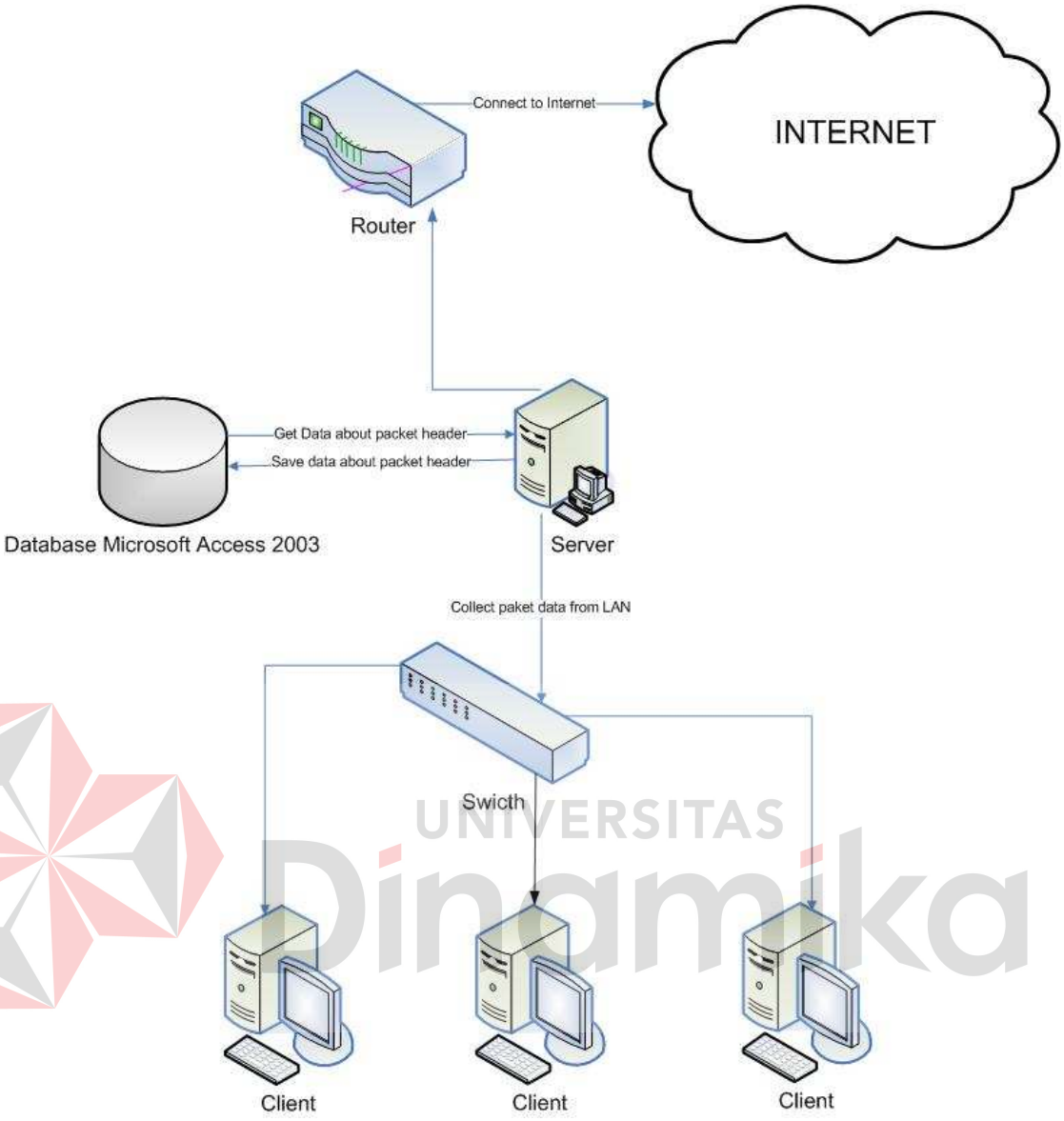
menunjukan bahwa paket tersebut adalah milik Skype dengan panduan literatur yang ada. Data yang sudah difilter akan disimpan dalam database server yang nantinya akan digunakan sebagai laporan kepada pengelola jaringan agar mengetahui *user* Skype yang sedang menggunakan fitur VoIP beserta dengan diagram paket datanya. Desain *input-output* dari proses ini adalah sebagai berikut:



Gambar 3.2. *Blok Diagram* untuk Identifikasi Pengguna Skype

Sehingga dengan adanya aplikasi ini diharapkan dapat mengetahui karakteristik VoIP pada Skype mulai dari paket byte, *destination* IP, *destination* Port dan *source* Portnya. Hasil yang didapat dari aplikasi ini adalah dalam satu jaringan lokal (LAN) seorang admin jaringan dapat mengetahui mana *user* yang sedang menggunakan Skype (baik yang sedang memakai fasilitas VoIP maupun yang tidak) dan *user* mana yang tidak menggunakan Skype.

Secara garis besar rancangan arsitektur sistem *hardware* yang akan dibangun dapat diketahui pada gambar 3.3 diagram topologi jaringan.



Gambar 3.3 Diagram Topologi Jaringan

Untuk mengimplmentasikan aplikasi ini dilakukan dengan menggunakan dua titik pengamatan yaitu:

1. Pengamatan penggunaan aktivitas Skype pada masing-masing komputer *client*
2. Akses internet ke Skype

Sebagai parameter untuk mengidentifikasi paket data Skype baik pada saat login maupun pada saat melakukan komunikasi VoIP, penulis bertumpu pada dua literatur yaitu literatur DongYan dkk. dan literatur Marcell Parenyi. Untuk

proses login Skype, dalam literatur Dongyan Zhang dkk., dia mendefinisikan IP address dari webserver ui.skype.com adalah sebagai berikut :

a. 193.88.6.228
b. 212.187.172.228
c. 212.72.49.136
d. 217.159.236.228

Menurut literatur yang sama, Skype menggunakan TCP connection port 80 untuk akses HTTP dan port 443 untuk akses HTTPS. DongYan juga memaparkan bahwa besarnya paket byte saat login adalah antara 25 byte sampai dengan 39 byte. Sedangkan untuk komunikasi VoIP, DongYan menjelaskan bahwa besar Packet_Byte nya berkisar antara 80 Byte hingga 640 Byte. Dan Port bebas yg digunakan Skype untuk komunikasi VoIP, menurut literatur Marcell Parenyi, adalah lebih dari 1024. Parameter-parameter ini yang akan digunakan sebagai bahan analisa ketika hasil pengujian telah selesai dilaksanakan.

3.2. Perancangan Sistem

Dalam proses pengembangan sebuah sistem dibutuhkan perencanaan terlebih dahulu. Hal ini bertujuan agar sistem atau aplikasi yang dibuat dapat berfungsi dengan baik (sesuai dengan yang diharapkan) yaitu dapat melakukan identifikasi lalu-lintas data VoIP Skype.

Dalam pengembangan aplikasi identifikasi lalu-lintas data VoIP Skype ini digunakan pemodelan *Unified Modelling Language* (UML). UML digunakan untuk menggambarkan sistem dari beberapa aspek untuk mendapatkan

pemahaman yang utuh terhadap sistem yang dibangun, dalam hal ini adalah sistem aplikasi identifikasi lalu-lintas data VoIP pada Skype.

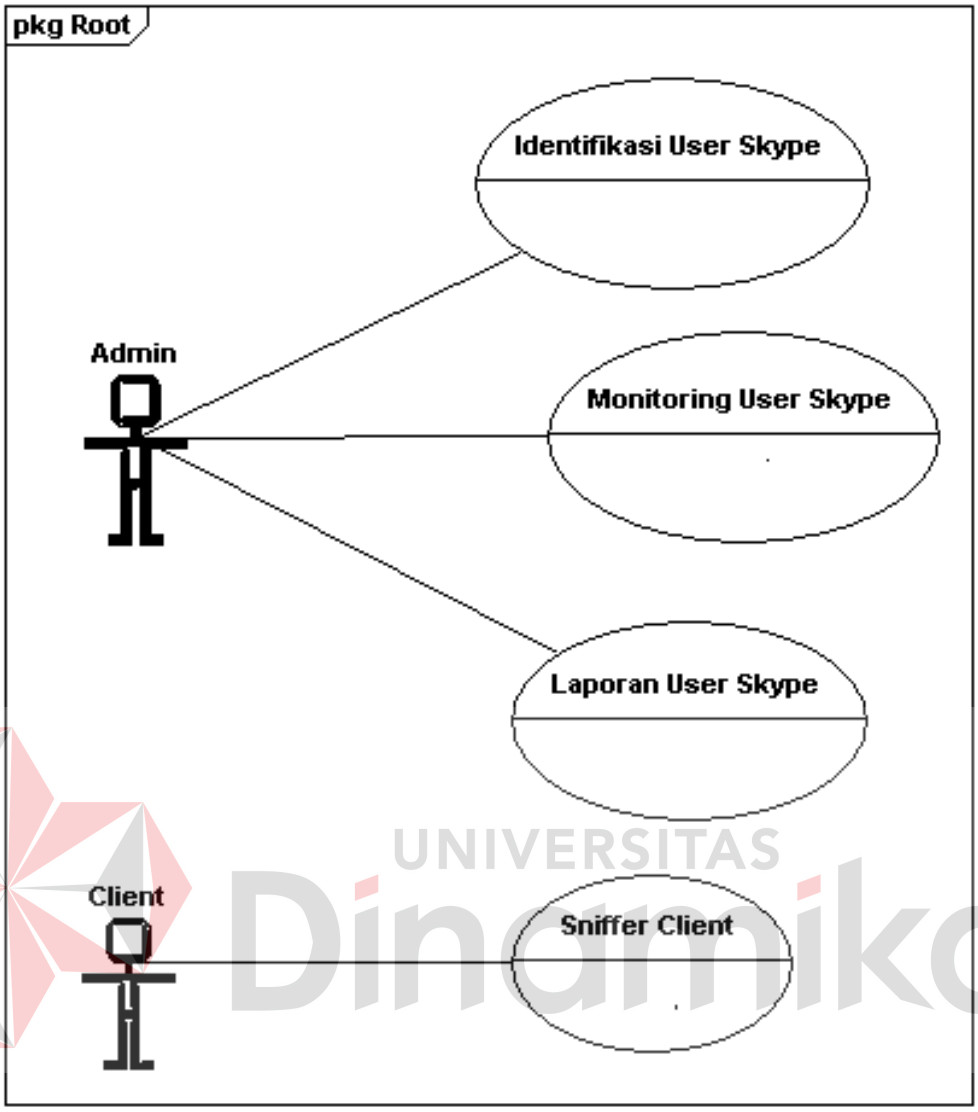
Dalam perancangan sistem ini terdapat beberapa tahapan pengembangan yang harus dilakukan dengan tujuan agar sistem yang dirancang menjadi lebih mudah untuk dibangun. Diagram yang dibutuhkan adalah *use-case* diagram, *activity* diagram, *sequence* diagram, dan yang terakhir adalah *class* diagram.

1. Diagram *Use-case*, adalah diagram yang bertujuan untuk mendokumentasikan beberapa aktor (yang ada di luar sistem), beberapa *use-case* (yang ada di luar sistem), dan hubungan antar mereka. Dengan kata lain, *use-case* menggambarkan bagaimana aktor menggunakan sistem.
2. Diagram *Activity*, menggambarkan aliran kejadian untuk masing-masing *use case*.
3. Diagram *Sequence*, menggambarkn interaksi yang disusun berdasarkan urutan waktu.

Masing-masing diagram di atas pada aplikasi identifikasi *user* Skype ini dibahas secara lengkap di bawah ini.

3.2.1. *Use-Case* Diagram Aplikasi Identifikasi Lalu-Lintas Data VoIP Skype

Use-case menggambarkan modul dari sistem atau persyaratan-persyaratan pada sistem dari sudut pandang pengguna sistem. Berikut adalah gambar *use-case* diagram pada aplikasi identifikasi lalu-lintas data VoIP Skype (pada *use-case* diagram hanya menggambarkan proses yang dilakukan secara elektronik).



Gambar 3.4 *Use-Case* Diagram Aplikasi Identifikasi Lalu-Lintas Data VoIP Skype

Pada gambar 3.4 terdapat dua aktor, yaitu: *administrator* dan *Client*. Selain terdapat dua aktor sebagai pengguna sistem, sistem ini juga terdiri dari 4 buah *use-case*, yaitu:

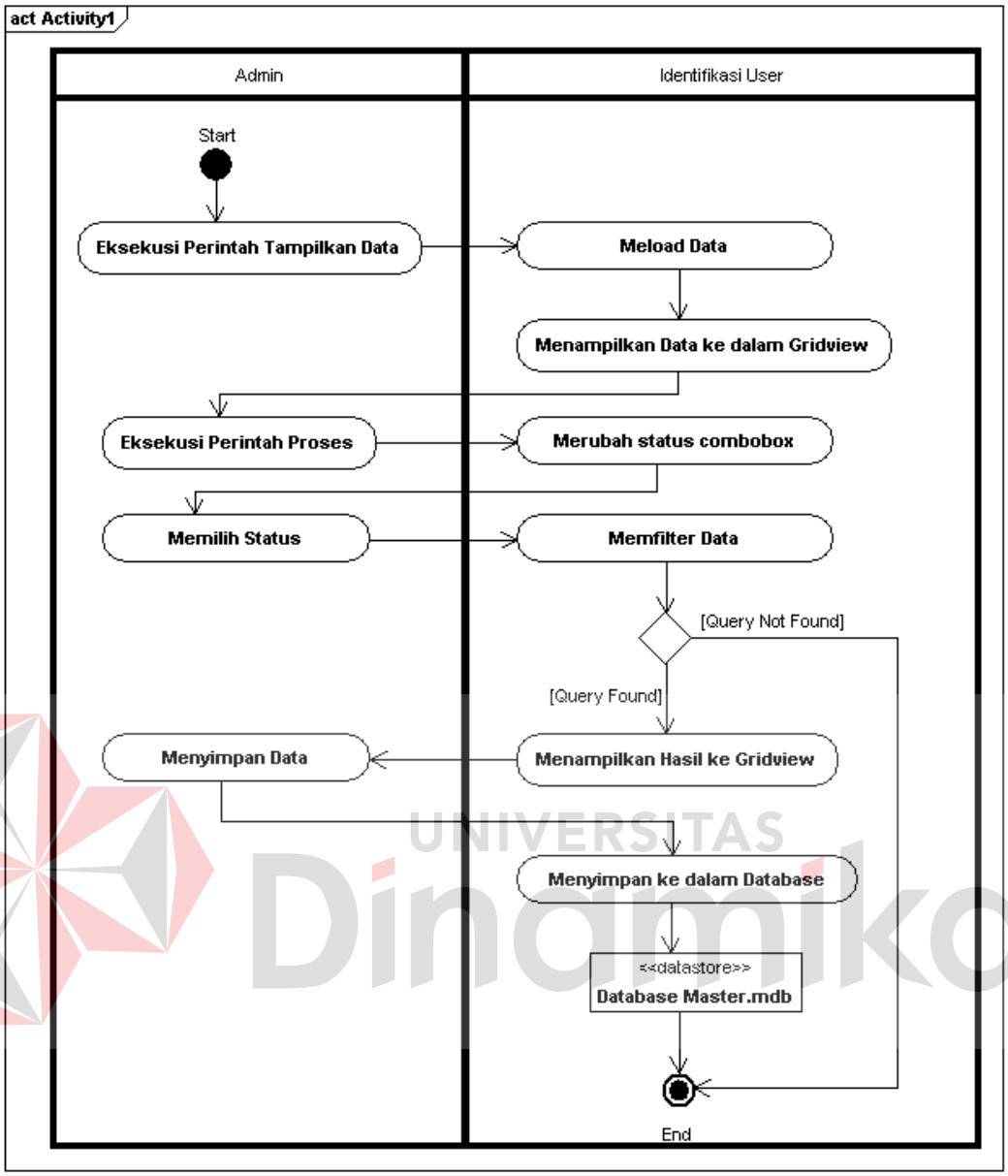
1. Identifikasi *User* Skype, pada *use-case* ini memungkinkan seorang aktor (Admin) untuk mencari lokasi database disimpan dan me*load* ke dalam aplikasi, kemudian menampilkan database hasil *sniffing* ke dalam DataGridView pada *form* identifikasi *traffic user*, memfilter paket data yang

diperoleh dari Client, dan memilih serta menyimpan status database manakah yang akan ditampilkan dan disimpan (paket data Skype *user* login atau VoIP).

2. *Monitoring User* Skype, pada use-case ini memungkinkan seorang aktor (Admin) untuk melihat alamat IP, komputer mana yang hidup dan sedang menggunakan Skype serta yang tidak. Admin juga dapat mengetahui user mana yang sedang menggunakan fasilitas VoIP Skype dan yang tidak.
3. Laporan *User* Skype, pada *use-case* ini memungkinkan seorang aktor (Admin) untuk menampilkan informasi pengguna Skype berdasarkan hasil *filtering* data yang diperoleh dari *Client.*
4. Sniffer Client, pada *use-case* ini memungkinkan seorang aktor (Client) untuk menentukan jaringan yang digunakan secara dinamis, memberikan batas konfigurasi waktu untuk proses *sniffing*, dan memulai proses *capture* data Client.

3.2.2. Activity Diagram dan Sequence Diagram Identifikasi *User* Skype

Use-case identifikasi *user* pengguna Skype adalah untuk memfilter paket data hasil *sniffing* yang diperoleh dari Client, untuk kemudian diidentifikasi yang merupakan paket data Skype dan yang bukan Skype. Pada *use-case* identifikasi *user* Skype terdapat dua buah diagram yang dapat menjelaskan proses yang dilakukan pada proses identifikasi *user* pengguna Skype tersebut. kedua diagram tersebut adalah *activity diagram* (diagram aktifitas) dan *sequence diagram*.



Gambar 3.5 *Activity* Diagram Identifikasi *User*

Pada diagram aktifitas di atas (gambar 3.5) terdapat dua buah kolom yang memisahkan atau kegiatan yang harus dilakukan oleh aktor maupun sistem. Pada diagram diatas kolom pertama digunakan oleh aktor yaitu admin. Sedangkan kolom kedua dipergunakan oleh sistem identifikasi *user*.

Pada diagram aktifitas ini terjadi proses pemilihan database oleh admin. Database yang dimaksud adalah database yang diperoleh dari proses *capturing* data pada Client. Admin akan memanggil database masuk ke dalam aplikasi,

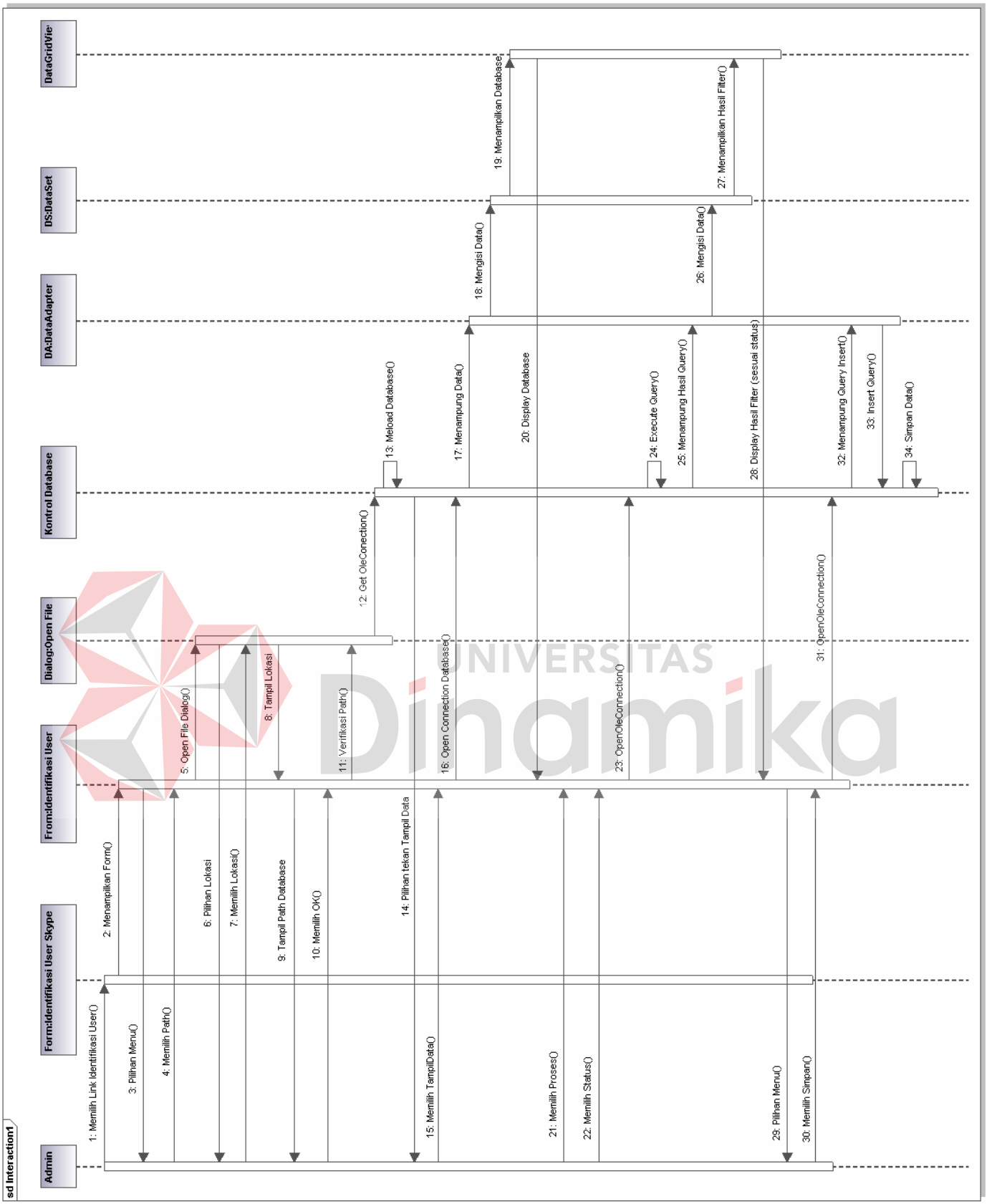
pertama-tama mengeksekusi perintah *path* untuk mencari lokasi database. Kemudian sistem akan menampilkan kotak dialog *Open File*. Aktor memilih dimana lokasi database yang akan difilter, sehingga sistem akan menampilkan alamat lokasi pada *path* database. Untuk me*load* data, aktor memberi perintah OK dan sistem akan berkoneksi dengan database serta me*load* database. Database yang telah di*load* akan ditampilkan pada DataGridView dalam sistem.

Setelah proses *load* database selesai, maka langkah berikutnya adalah *filtering* paket. Seperti yang terlihat pada gambar 3.5 *activity* diagram *filtering* paket di atas, 8 buah proses selanjutnya yang dimulai dari proses admin menekan perintah proses, pada perintah proses ini memberikan penjelasan bahwa setelah mengeksekusi perintah proses kemudian status pada combobox telah dipilih maka sistem akan memfilter paket. Untuk status login, *filtering* dilakukan berdasarkan Port (80.443) dan *Destination* IP (194.192.199.251, 194.165.188.101, 212.187.17.78, 204.9.163.158). Kemudian untuk status VoIP, dasar *filtering* paket sama dengan dasar *filtering* paket untuk login namun ditambahkan besaran paket size. Karena dari besaran paket size inilah dapat terlihat perbedaan antara *user* yang sedang menggunakan fitur VoIP dan yang tidak. Jika *query* tidak ditemukan, maka aktivitas berakhir. Namun jika *query* ditemukan, sistem akan menampilkannya ke dalam datagridview. Selanjutnya admin mengeksekusi perintah untuk menyimpan data hasil *query*, dan sistem kemudian menyimpan hasil ke database yang baru.

Setelah melakukan proses penggambaran diagram aktifitas, langkah berikutnya adalah menentukan kelas; metoda-metoda dan atribut yang dibutuhkan

untuk meciptakan aplikasi yang dibutuhkan pada sub sistem identifikasi *user*. Gambar diagram *sequence* identifikasi *user* dapat dilhat pada gambar 3.6.

Pada gambar *sequence* diagram identifikasi *user* (gambar 3.6), sistem dijalankan oleh satu aktor yaitu Admin. Pada langkah awal, aktor mengekseskusi halaman identifikasi *user* dari *form* utama identifikasi *traffic*, kemudian sistem akan mengeksekusi beberapa kelas sesuai dengan nomor urut yang tertera pada gambar di atas. Pada saat halaman identifikasi *user* dieksekusi, maka halaman tersebut akan menampilkan beberapa menu yang belum *enable* dan sudah *enable*.



Gambar 3.6 *Sequence* Diagram Identifikasi *User*

Menu pertama yang *enable* ada perintah *path* pada tombol *path* untuk memilih lokasi database. Pada saat perintah ini dieksekusi oleh admin, sistem akan membuka *open file* dialog. Admin diharuskan untuk mencari dan memilih database yang ingin diproses. Kemudian setelah dieksekusi, sitem akan menampilkan alamat lokasi database pada *path* database. Kemudian saat admin mengeksekusi perintah OK, sistem akan memverifikasi lokasi dari *path* database, kemudian mengkoneksikan diri dengan database melalui ole*Connection* database. Sistem akan me*load* database dan menampilkannya pada dataGridView saat aktor (admin) mengeksekusi perintah TampilData.

Proses tampilnya data ke datagrid ini dimulai dengan dibukanya koneksi dengan database, kemudian seluruh data yang ada pada database ditampung oleh kelas DataAdapter. Kelas DataAdapter inilah yang akan membawa dan mengisikan data ke DataSet, sehingga data yang ada pada DataSet dapat ditampilkan pada DataGridView.

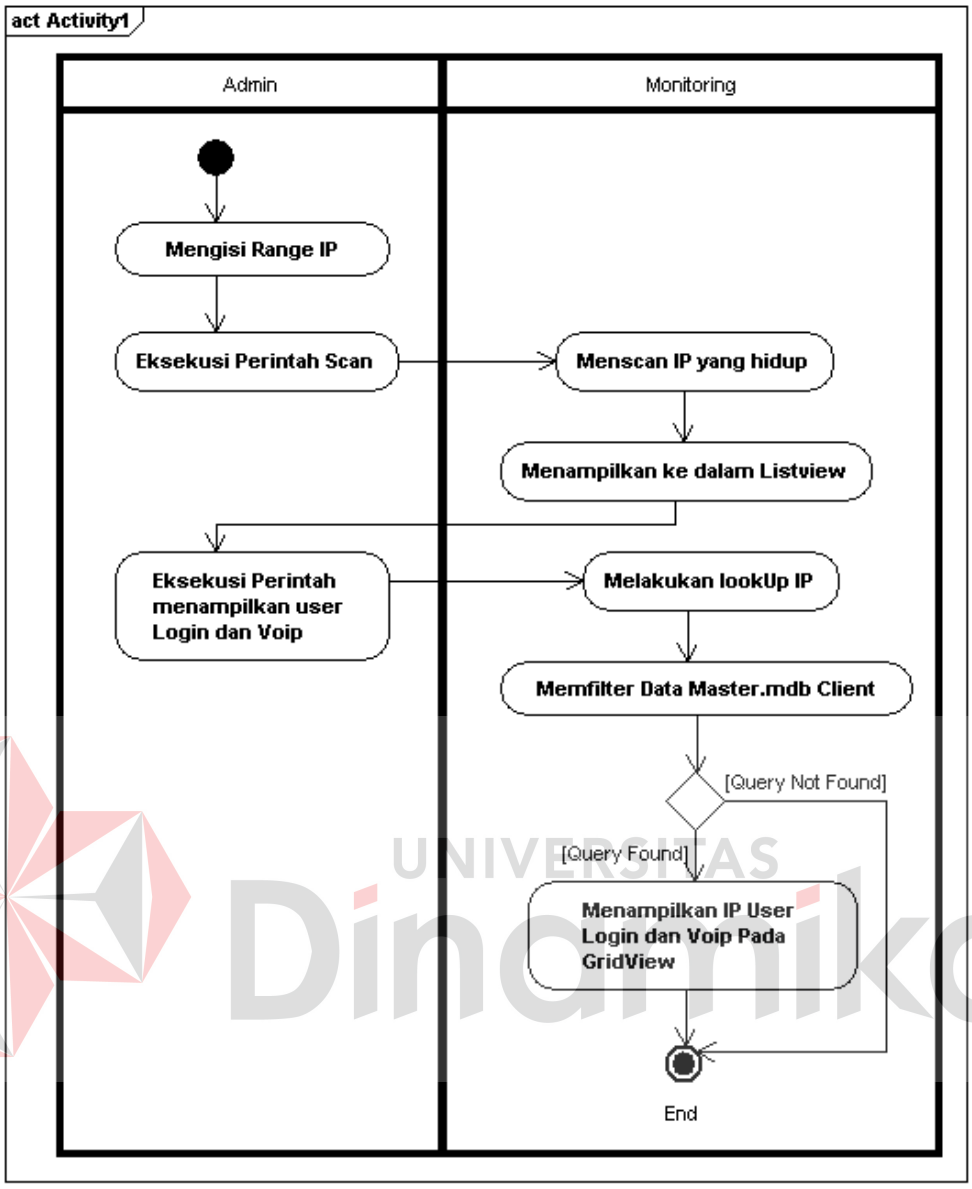
Langkah berikutnya adalah proses untuk mengidentifikasi *user*. Langkah ini merupakan kelanjutan dari langkah sebelumnya setelah database yang akan difilter berhasil ditampilkan pada DataGrid. Bermula dari hal tersebut, selanjutnya admin akan mengeksekusi perintah proses dan memilih status (login atau VoIP) pada combobox yang tersedia. Setelah status dipilih, sistem akan mengexecute *query* yang ada dalam database. Kemudian sistem memanggil kelas DataAdapter dan hasil dari *query* akan ditampung kembali oleh kelas DataAdapter ini. Kelas ini pulalah yang akan kembali mengisi data ke kelas DataSet untuk kemudian ditampilkan pada DataGridView.

Untuk menyimpan hasil filter ini, admin mengeksekusi perintah Simpan, sehingga sistem mengeksekusi *Open* ole*Connection* database yang akan mengontrol kelas DataAdapter untuk membawa data hasil filter serta menginsertkan ke dalam database yang baru. Hasil *query* ini kemudian disimpan oleh database.

3.2.3. Activity Diagram dan Sequence Diagram *Monitoring* Skype

Use-case monitoring Skype adalah untuk melihat secara IP, komputer mana yang sedang menggunakan Skype dan yang tidak. Admin juga dapat mengetahui *user* mana yang sedang menggunakan fasilitas VoIP Skype dan yang tidak. Pada *use-case* monitoring Skype terdapat dua buah diagram yang dapat menjelaskan proses yang dilakukan pada proses monitoring Skype. Kedua diagram tersebut adalah *activity diagram* (diagram aktifitas) dan *sequence diagram*.

Pada gambar *activity* diagram *monitoring* Skype (gambar 3.7), langkah awal adalah mengisi *range* IP yang dilakukan oleh admin. Selanjutnya admin mengeksekusi perintah *scanning* data, sehingga sistem melakukan aktivitas *scanning* IP *user* yang sedang *online*. Selama proses scanning akan ditampilka tulisan >>*Please wait while processing is done*<<. Jika proses scanning telah menemukan IP komputer yang menyala, maka hasil *scanning* ini ditampilkan oleh sistem ke dalam ListView. Ketika proses scanning selesai, maka sistem akan menampikan tulisan "*All done search retrieved ... working stations*", dimana titik-titik berisi jumlah IP client yang berhasil di*scanning*.



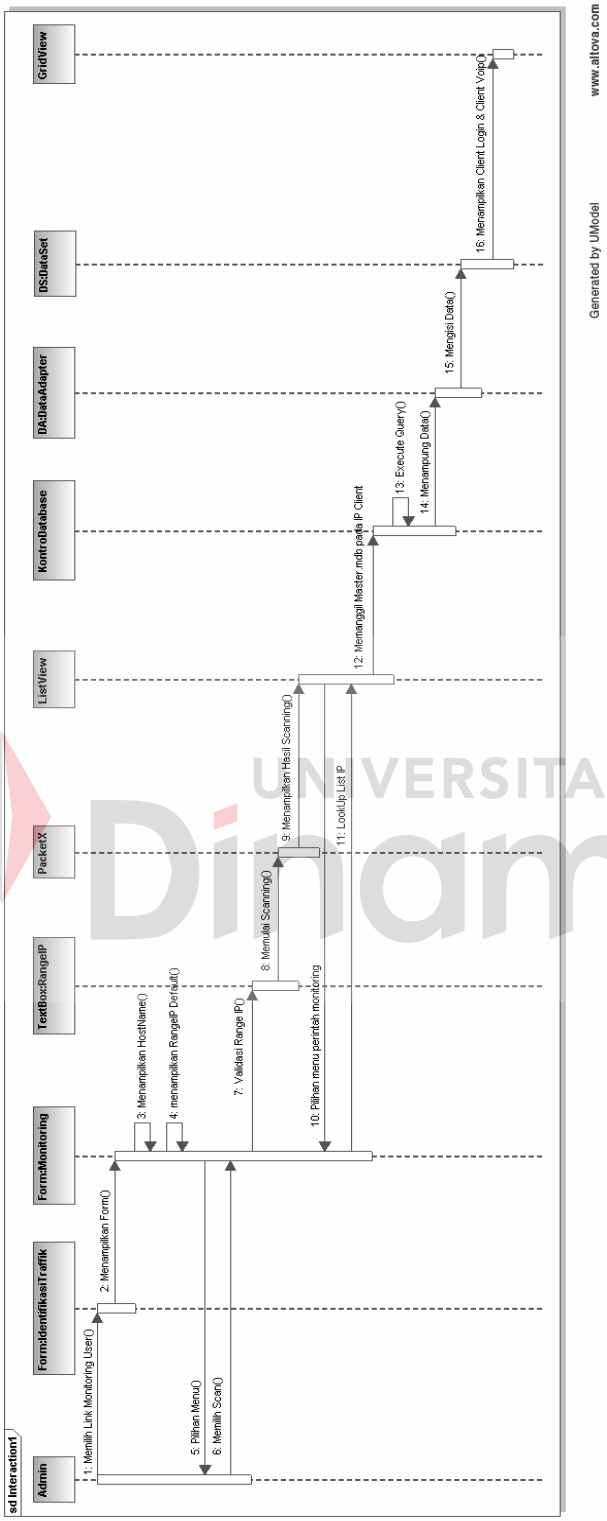
Gambar 3.7 *Activity* Diagram *Monitoring* Skype

Setelah proses scanning, aktivitas selanjutnya adalah menampilkan IP user yang login dan yang VoIP. Pada saat perintah ini dieksekusi, sistem akan mulai meng-*query* data dari database hasil filter sehingga akan diketaui IP mana yang menunjukkan bahwa penggunanya sudah Login Skype dan yang sedang melakukan aktivitas VoIP.

Setelah melakukan proses penggambaran diagram aktifitas, langkah berikutnya adalah menentukan kelas; metoda-metoda dan atribut yang dibutuhkan untuk meciptakan aplikasi yang dibutuhkan pada sub sistem *monitoring* Skype.

Pada gambar 3.8 (*Sequence* Diagram *Monitoring* Skype) terdapat satu aktor yaitu admin yang akan menjalankan sub sistem ini. Pada langkah awalnya, aktor memilih *link monitoring user* pada *form* utama, kemudian *form* utama akan memanggil kelas *monitoring user*. Aktor kemudian memberikan isian *range* IP pada *form monitoring user*. Setelah itu aktor memberikan perintah *scan* untuk memulai proses *scanning* jaringan LAN.

Pada saat perintah *scan* dieksekusi, sistem memvalidasi isian *range* IP yang terdapat pada *textbox Range* IP. Setelah itu sistem memulai *scanning* dengan dipanggilnya kelas PaketX. Hasil dari proses *scanning* ini akan ditampilkan di ListView. Kemudian aktor mengeksekusi perintah untuk menampilkan client yang *Login* dan VoIP melalui tombol *Client Login* dan *Client* VoIP. Pada saat perintah dieksekusi, aplikasi melakukan *LookUp* IP pada ListView untuk memanggil databasae master.mdb pada masing-masing client. Sistem kemudian mengkoneksikan diri dengan database dan meng*execute query* yang ada dalam database. Kemudian sistem memanggil kelas DataAdapter dan hasil dari *query* akan ditampung kembali oleh kelas DataAdapter ini. Kelas ini pulalah yang akan mengisi data ke kelas DataSet untuk kemudian ditampilkan pada DataGridView. Tampilan data hanya berupa IP saja.

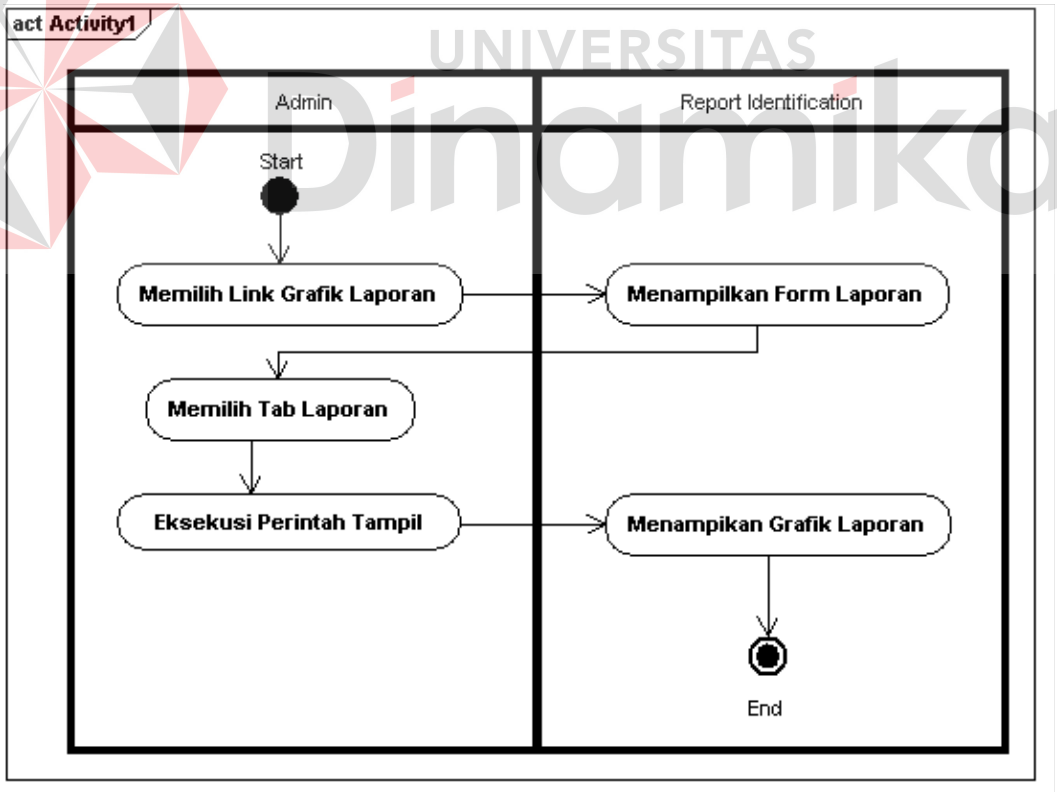


Gambar 3.8 *Sequence* Diagram *Monitoring* Skype

3.2.4. Activity Diagram dan Sequence Diagram Laporan *User* Skype

Use-Case laporan *user* Skype adalah *use-case* yang dipergunakan untuk menampilkan informasi pengguna Skype berdasarkan hasil *filtering* data pada form identifikasi *user* Skype. Pada *use-case* laporan *user* Skype akan dipergunakan sebagai kelanjutan dari proses identifikasi *traffic*. Sama dengan sebelumnya, pada *use-case* laporan *user* Skype terdapat dua buah diagram yang menjelaskan proses yang dilakukan yaitu *activity diagram* (diagram aktifitas) dan *sequence diagram*.

Pada *activity* diagram pelaporan pengguna Skype ini akan menampilkan informasi grafik laporan pengguna Skype dari hasil *filtering* data pada *use-case* identifikasi *user* Skype.



Gambar 3.9 *Activity* Diagram Pelaporan *User* Skype

Pada gambar 3.9 *activity* diagram (diagram aktifitas) pelaporan *user* Skype terdapat 5 (lima) buah proses. Setelah proses filter data pada *use-case* identifikasi pengguna Skype dilakukan, hasil *filtering* akan disimpan ke dalam database. Data yang tersimpan itulah yang dijadikan sebagai bahan pelaporan. Pertama-tama admin memilih *link* Grafik Laporan pada halaman utama terlebih dahulu. Kemudian sistem akan menampilkan *form* antarmuka *report identification* yang terdiri dari tiga tab, yaitu laporan pengguna, laporan rata-rata paket, dan laporan VoIP. Selanjutnya admin memilih salah satu tab yang dikehendaki. Kemudian pada antarmuka tab yang telah dipilih, admin mengeksekusi perintah "Tampil" untuk memerintahkan sistem melakukan proses pelaporan, sehingga sistem akan melakukan proses menampilkan diagram laporan atau diagram *Crystal Report*. Setelah sistem melakukan proses pelaporan pengguna Skype dalam bentuk diagram *Crystal Report* tadi, maka aktivitas pelaporan pengguna Skype berakhir.

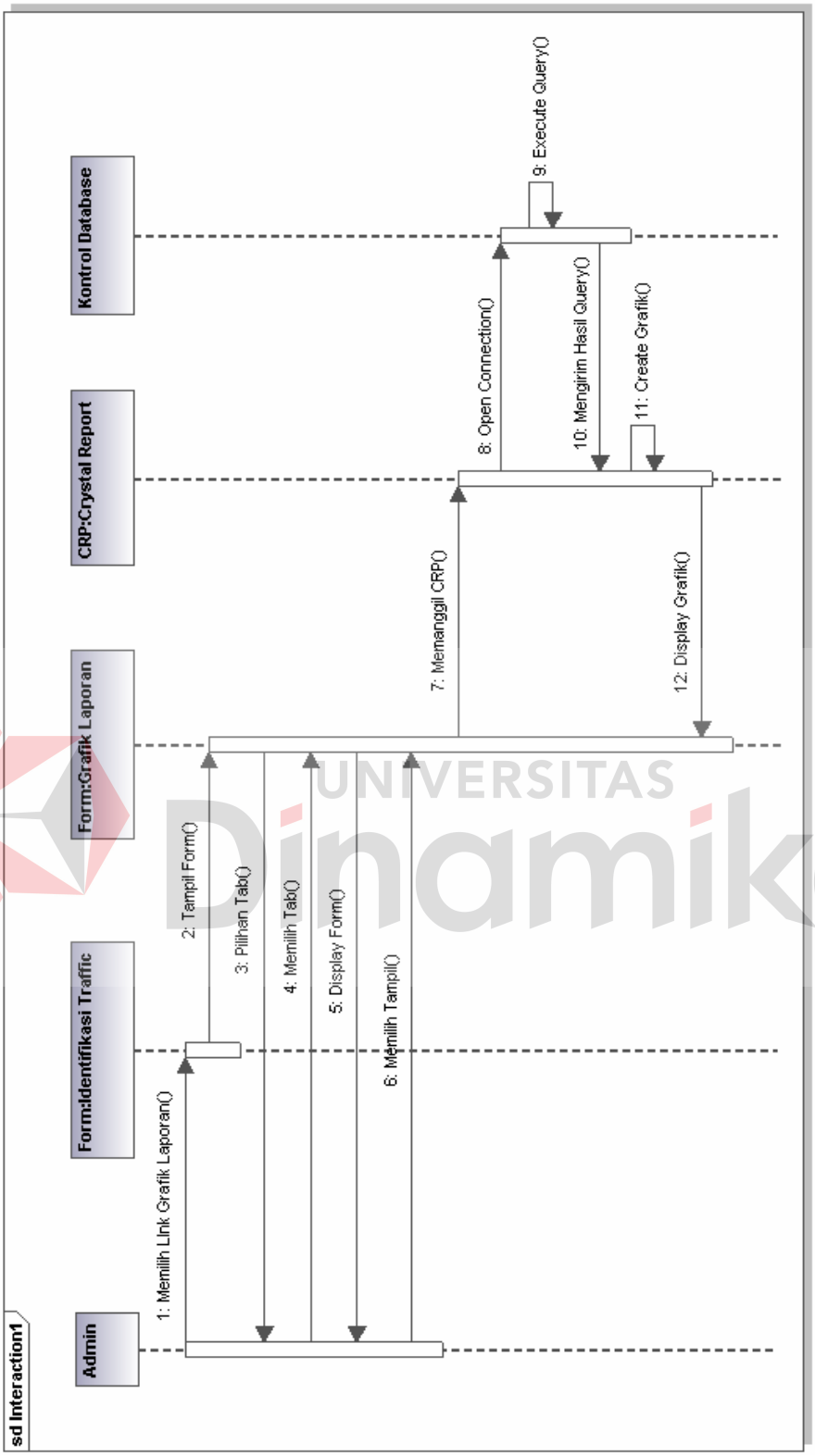
Setelah proses perancangan aktifitas yang dibutuhkan pada *use-case* laporan *user* Skype, maka langkah selanjutnya adalah penyusunan *sequence* diagram (diagram sekuensial). Tujuan dibuatnya diagram sequensial adalah untuk mempermudah proses pembuatan maupun perubahan kode program (bila terjadi perubahan maupun penambahan fasilitas atau attribut) aplikasi. Berikut adalah diagram sekuensial untuk proses pelaporan pengguna Skype.

Pada gambar *sequence* diagram pelaporan *user* Skype (gambar 3.10) terdapat satu aktor yaitu admin yang menjalankan sub sistem laporan *user* Skype ini.

Pada langkah awal sub sistem ini, aktor memanggil kelas grafik laporan, kemudian sistem akan menampilkan halaman tersebut untuk dilakukan proses

pemilihan tab laporan. Setelah tab dipilih oleh aktor, maka tahap berikutnya adalah pemberian perintah untuk menampilkan digram laporan dengan cara menekan tombol Tampil. Setelah aktor menekan tombol tampil, maka sistem akan memanggil kelas CRP:*CrystalReport*.

Kelas CRP:*CrystalReport* akan membuka koneksi dengan databse hasil filter, dan mengexecute *query* yang ada di dalamnya. Kemudian kelas kontrol database akan mengirimkan hasil *query* ke kelas CRP:*CrystalReport*. Dari sinilah, kelas CRP:*CrystalReport* mengeksekusi grafik. Selanjutnya kelas CRP:*CrystalReport* lah yang menampilkan diagram ke antarmuka Grafik laporan.

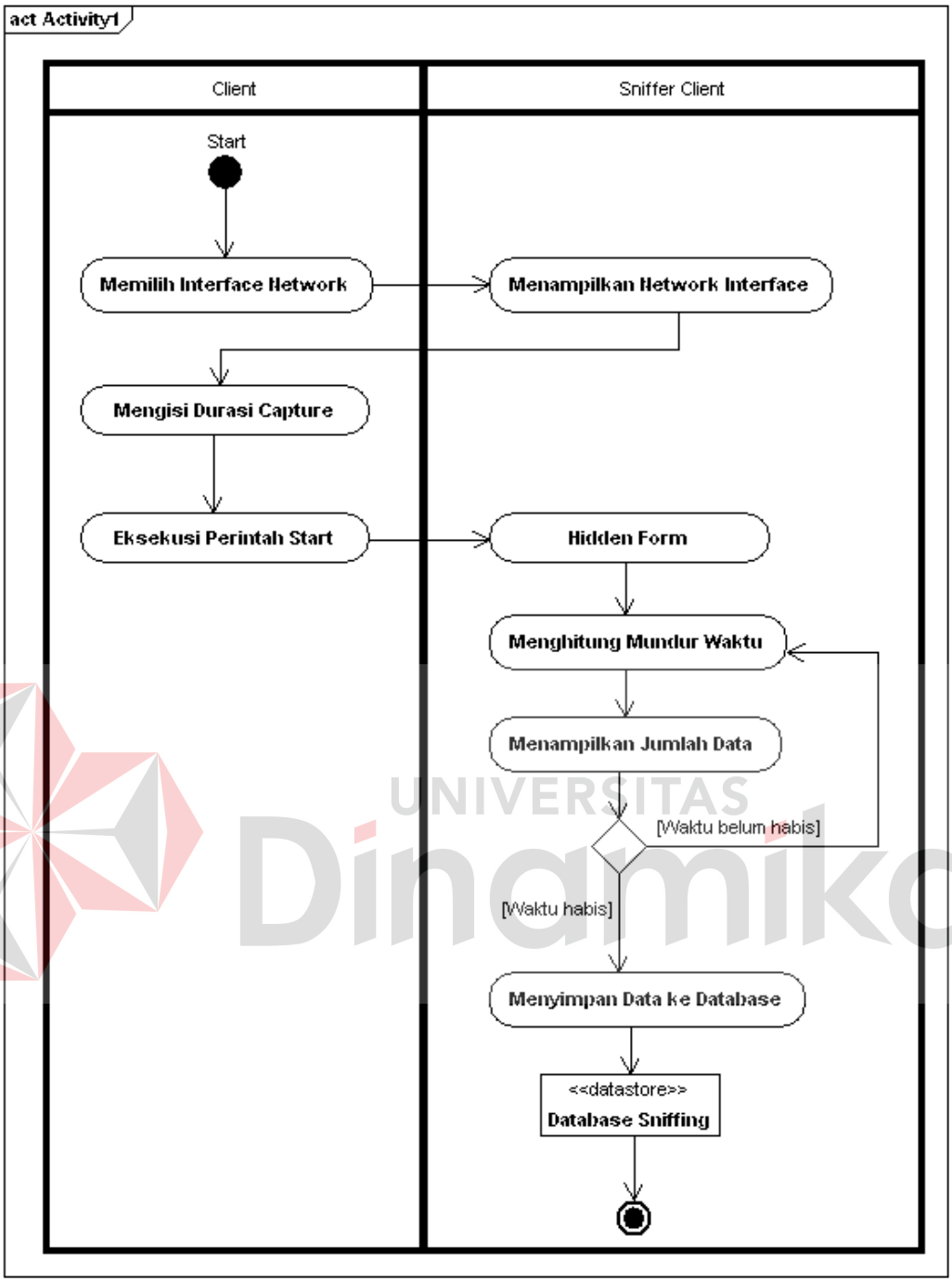


Gambar 3.10 *Sequence* Diagram Laporan *User* Skype

3.2.5. Activity Diagram dan Sequence Diagram Sniffer Client

Pada saat akan melakukan proses *sniffing*, client diharuskan untuk menentukan dahulu *network interface card* yang digunakan. Pilihan *network interface card* tergantung konfigurasi yang digunakan oleh PC Client. Proses *sniffing* juga dibatasi waktu sesuai pengaturan durasi waktunya. Proses ini berfungsi sebagai langkah awal untuk melakukan *capturing data*, dimana data yang tersimpan nantinya dijadikan bahan untuk pemfilteran selanjutnya oleh pihak Admin.

Pada sniffer client terdapat dua jenis diagram, yaitu *activity* diagram dan *sequence* diagram. Berikut adalah *activity* diagram proses sniffer client pada aplikasi identifikasi lalu-lintas data VoIP Skype ini.

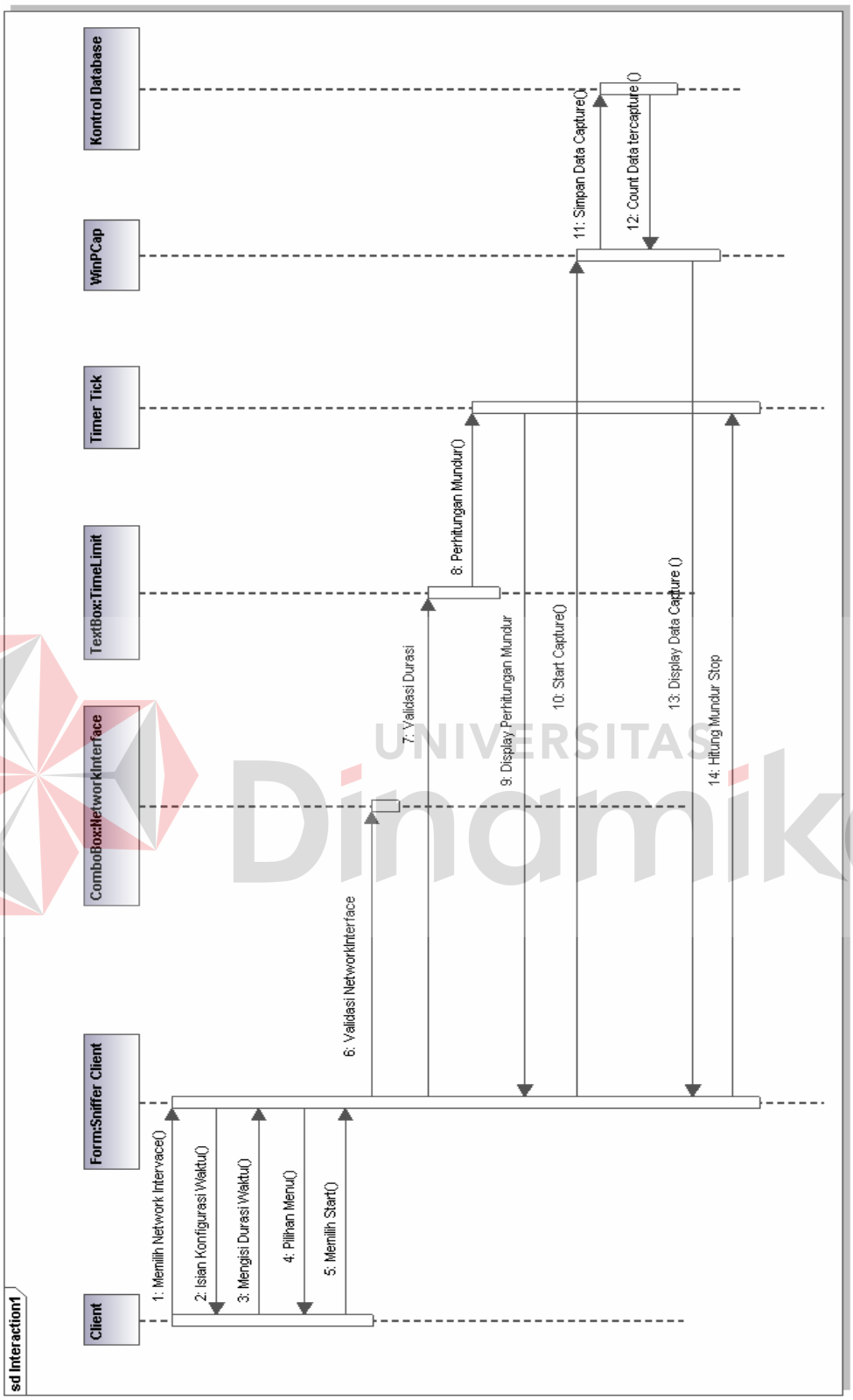


Gambar 3.11 *Activity* Diagram Sniffer Client

Pada gambar 3.11 terlihat 8 (delapan buah) buah aktifitas dengan proses kali pertama dilakukan oleh aktor (yaitu client) dengan memilih *interface* jaringan yang digunakan. Pemilihan *interface* jaringan ini disesuaikan dengan konfigurasi komputer client yang digunakan. Aktifitas berikutnya adalah mengisikan lama

waktu atau durasi waktu yang dikehendaki client untuk melakukan proses *sniffing*, dengan tujuan agar lamanya waktu *capturing* data terkontrol oleh client. Aktifitas terakhir yang dilakukan oleh aktor adalah mengeksekusi perintah *start*, sehingga proses *sniffing* dapat berlangsung. Lamanya waktu yang digunakan oleh sistem untuk proses tergantung pada isian durasi waktu yang diisikan oleh client pada aktifitas awal tadi. Pada saat proses *capturing* dimulai, sistem akan secara otomatis menyembunyikan diri atau *hidden*. Selanjutnya sistem melakukan aktivitas hitung mundur sampai durasi menyatakan 0 (nol) detik atau berakhir. Setelah proses *sniffing* selesai, maka sistem akan menampilkan jumlah data yang ter*capture* pada label *The Number of Data*, kemudian data hasil *sniffing* disimpan ke dalam sebuah database secara otomatis pada komputer client.

Dari proses penentuan aktivitas dasar yang harus dilakukan oleh pengguna, maka langkah selanjutnya adalah proses penggambaran *sequence* diagram sniffer client. Berikutnya adalah gambar *sequence* diagram pada *use-case* sniffer client yang dapat dilihat pada gambar 3.12.



Gambar 3.12 *Sequence* Diagram Sniffer Client

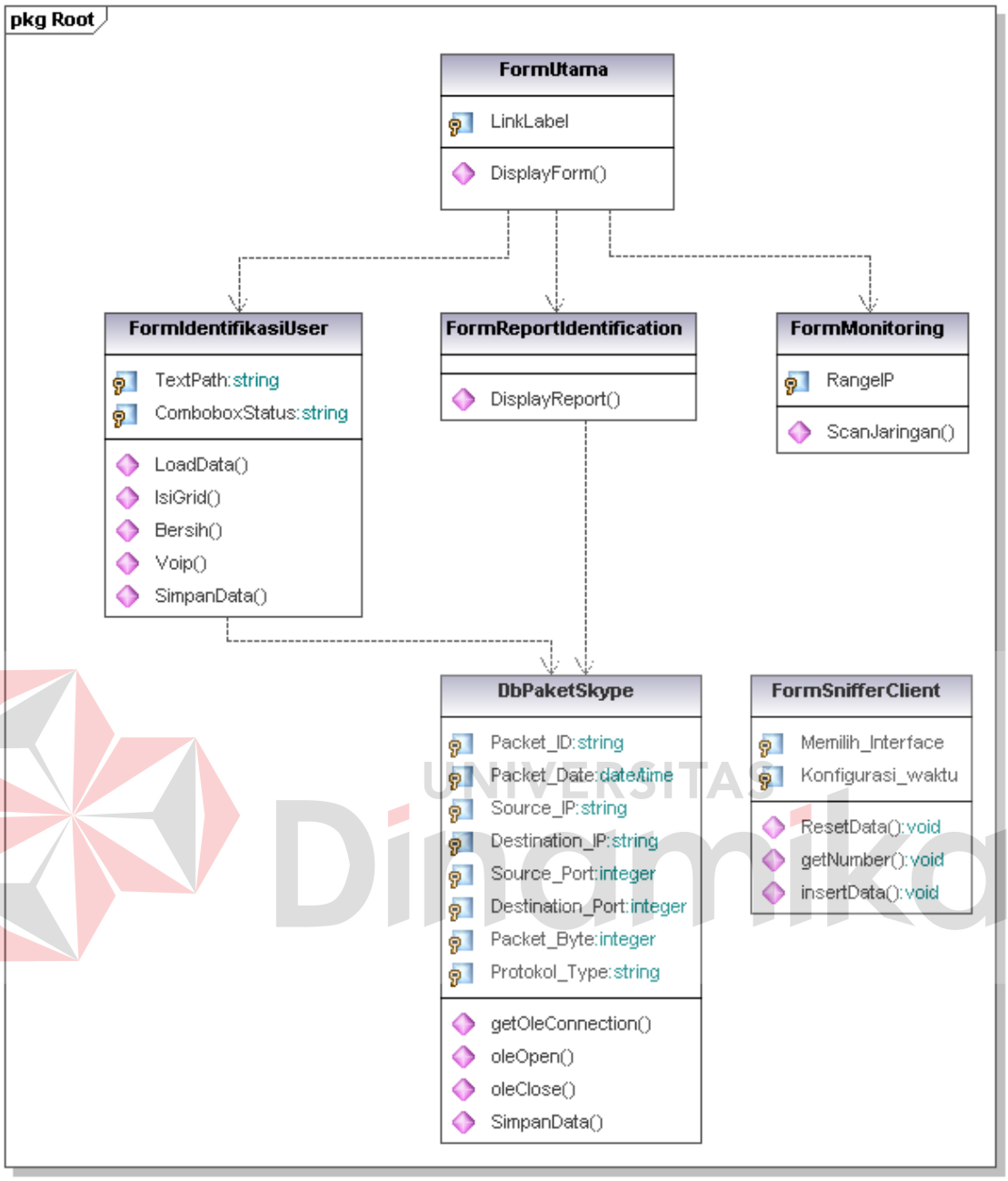
Pada *sequence* diagram sniffer client (Gambar 3.12), terdapat satu aktor yaitu client yang akan menjalankan sub sistem ini. Pada langkah awalnya, aktor memanggil kelas sniffer client, memilih network *interface* dan mengisi isian durasi waktu yang dikehendaki untuk pelaksanaan proses *sniffing*. Setelah itu aktor memberikan perintah *start* untuk memulai proses *sniffing* pada komputer client. Pada saat perintah ini dieksekusi, sistem memanggil kelas combobox untuk memvalidasi network *interface* dan juga memanggil kelas *textbox* untuk membatasi durasi *capturing* data sesuai isian yang diberikan oleh aktor. Setelah itu, sistem akan menghitung mundur waktu dengan dipanggilnya kelas TimerTick. Perhitungan mundur ini didisplay ke *form* sniffer client.

Sejak perhitungan mundur dimulai, sistem di saat itu pula memulai *capturing* data dengan dipanggilnya kelas WinPCap untuk melaksanakan tugas tersebut. Kelas WinPCap akan memanggil kelas kontrol database untuk membuka koneksi ke database guna menyimpan hasil *capturing* secara langsung ke database. Dari data yang disimpan ke database inilah kelas WinPCap menghitung data dan menampilkan jumlah data yang ter*capture* pada antarmuka sniffer client. Proses *capturing* selesai pada saat kelas *TimerTick* selesai menghitung mundur. Setelah selesai, sistem memanggil kelas kontrol database untuk mendisplay hasil *capturing*. Terakhir, aktor menyimpan hasil *capturing*.

3.2.6. *Class* Diagram Identifikasi *Traffic* VoIP

Dalam teknik perancangan dan pembuatan sebuah pengidentifikasian VoIP pada jaringan dengan studi kasus Skype ini dibutuhkan sebuah desain yang dapat menggambarkan sistem yang dibutuhkan oleh pengguna sistem. Selain tujuan tersebut, dengan adanya proses mendesain ini pengembang dapat mengkomunikasikan sistem yang akan dibuat kepada pengguna sistem. Selain untuk mengetahui bentuk identifikasi VoIP yang diinginkan oleh pengguna dan para pembuat aplikasi dapat melakukan pengelompokan dari tiap-tiap fungsi proses yang dibuat dan memudahkan dalam proses memperbaiki kesalahan logika.

Dalam identifikasi lalu-lintas VoIP pada jaringan dengan studi kasus Skype ini terdapat empat lapisan program yang dibuat, yaitu sniffer client, identifikasi *user*, *monitoring user* dan laporan *user* VoIP Skype. Berikut adalah gambar *Class* Diagram aplikasi identifikasi VoIP pada Skype.



Gambar 3.13 *Class* Diagram Identifikasi *Traffic* VoIP Skype

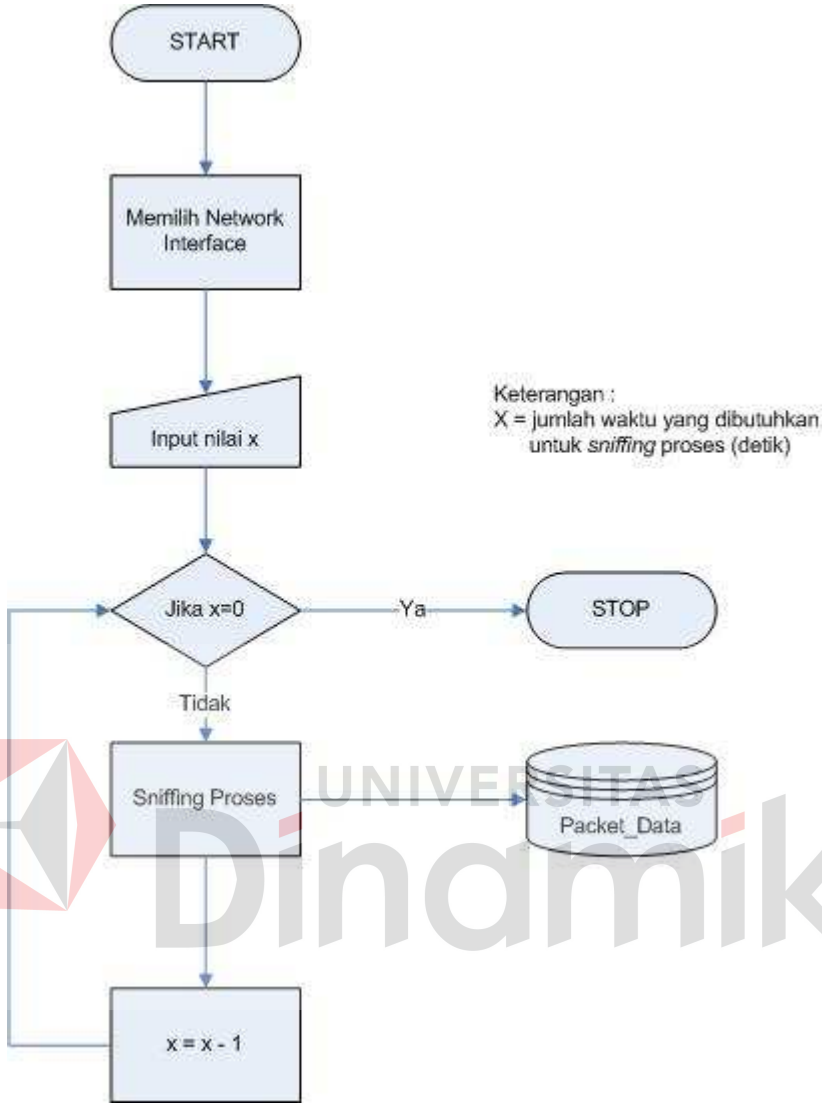
Pada gambar *class* diagram identifikasi *traffic* VoIP Skype (Gambar 3.13) terdapat 6 kelas, yaitu kelas *Form*Utama, *Form*Identifikasi*User*, *FormReportIdentification*, *FormMonitoring*, DbPaketSkype, dan *Form*SnifferClient. Dalam sistem, *Form*Utama menampilkan tiga kelas yaitu *Form*Identifikasi*User*, *FormMonitoring*, dan *FormReportIdentification*. Oleh

karena itulah antara kelas *Form*Utama, *Form*Identifikasi*User*, *FormMonitoring*, dan *FormReportIdentification* terdapat relasi dependensi dengan arah anak panah dari *Form*Utama menuju ke tiga kelas *form* di bawahnya. Arah anak panah menunjukkan bahwa kelas *Form*Utama memanggil kelas *Form*Identifikasi*User*, *FormMonitoring*, dan *FormReportIdentification*. Dengan relasi dependensi *Form*Utama tidak memiliki atribut instan bertipe *Form*Identifikasi*User* atau *FormMonitoring* atau *FormReportIdentification*.

Di dalam *Form*Identifikasi*User* terdapat satu metoda SimpanData() yang digunakan untuk menyimpan hasil *filtering* paket ke kelas DbPaketSkype. Dengan kata lain kelas *Form*Identifikasi*User* bergantung pada kelas DbPaketSkype, namun *Form*Identifikasi*User* tidak memiliki atribut instan bertipe kelas DbPaketSkype. Karena itulah kelas *Form*Identifikasi*User* juga berelasi dependensi dengan kelas DbPaketSkype.

Yang terakhir adalah kelas *FormReportIdentification* yang berelasi dependensi dengan kelas DbPaketSkype karena kelas *FormReportIdentification* menampilkan Display*Report* yang *query*nya bergantung pada kelas DbPaketSkype. Sedangkan kelas *Form*SnifferClient berdiri sendiri.

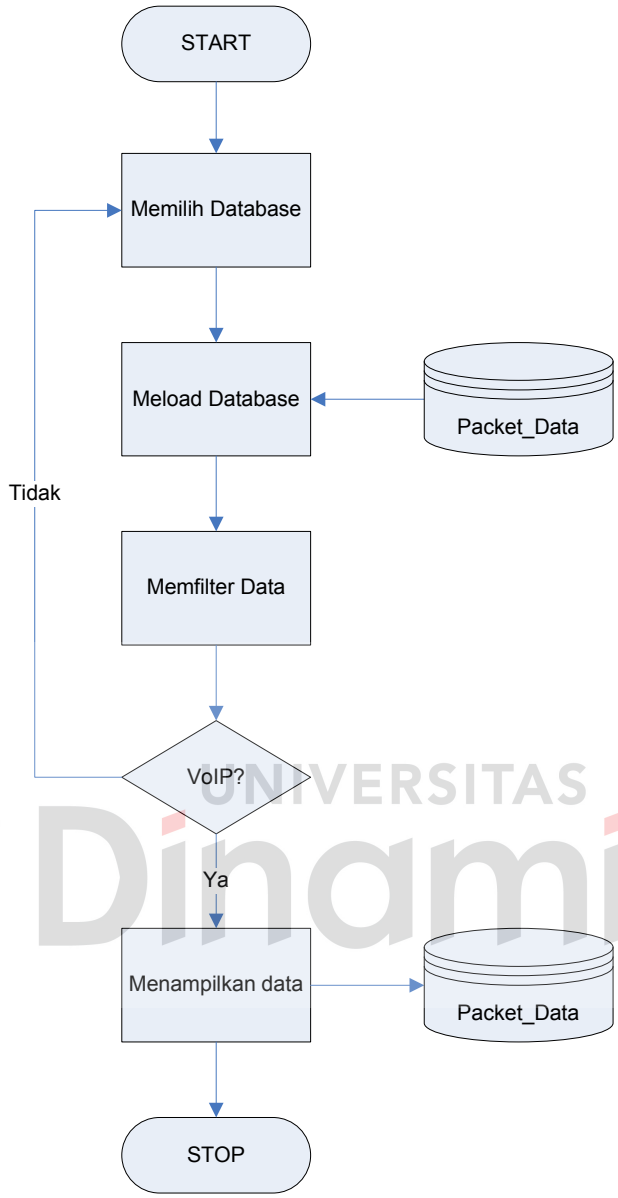
3.2.7. Flow Chart Sniffer Client



Gambar 3.14 *Flow Chart Sniffer Client*

Gambar 3.14 Desain umum *sniffing packet data* yang bermula dari melakukan *sniffing packet data* di setiap *host* kemudian data hasil *sniffing* disimpan kedalam *database sniffer* yang selanjutkan akan diteruskan kedalam proses identifikasi data Skype khususnya VoIP.

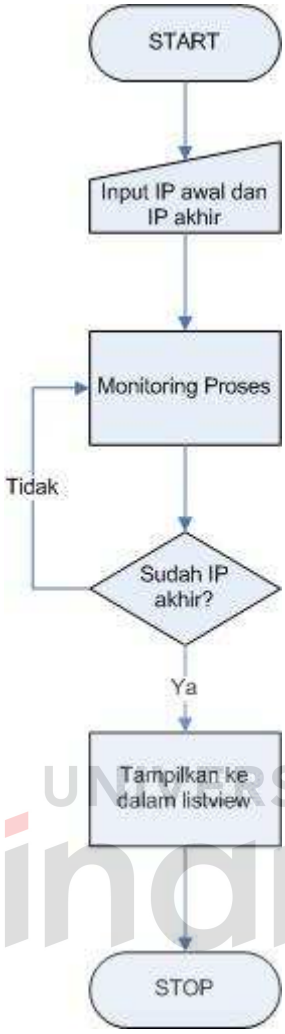
3.2.8. Flow Chart Identifikasi Traffic VoIP



Gambar 3.15 *Flow Chart* Identifikasi *Traffic* VoIP

Gambar 3.15 Desain umum Identifikasi *Traffic* VoIP menggambarkan alur proses identifikasi paket data yang bermula dari mengambil data di *database sniffer* milik *client* hingga memfilter data. Dan hasil akhirnya adalah tampilan paket data VoIP Skype.

3.2.9. Flow Chart Monitoring User VoIP



Gambar 3.16 *Flow Chart Monitoring User* VoIP

Gambar 3.16 Desain umum *monitoring user* VoIP menggambarkan alur proses monitoring user Skype yang bermula dari memasukkan IP awal dan IP akhir. Kemudian proses monitoring dimulai untuk mencari IP yang hidup dari IP awal hingga IP akhir. Selanjutnya menampilan ke dalam *ListView* beserta nama komputer dan status pada masing-masing *user*.

3.3. Struktur Fisik Database

Struktur tabel yang digunakan untuk menggambarkan secara detail tentang tabel-tabel yang digunakan dan disesuaikan dengan kebutuhan aplikasi ini. Tabel-tabel yang digunakan dalam aplikasi ini antara lain:

1. Nama Tabel : computer_data

   Fungsi : menyimpan data mengenai komputer

   Primary Key : IP_Address

   Foreign Key : -

   Struktur tabelnya adalah sebgaimana terlihat pada tabel 3.1 di bawah ini:

Table 3.1 Tabel computer_data

FIELD	***TYPE***	***LENGTH***	***CONSTRAINT***
IP_Address	varchar	15	*Primary Key*
Computer_name	varchar	30	
Time_Limit	number	4	

2. Nama Tabel : packet_data

   Fungsi : menyimpan mengenai *packet header*

   Primary Key : Packet_ID

   Foreign Key : -

   Struktur tabelnya adalah sebgaimana terlihat pada tabel 3.2 di bawah ini:

Table 3.2 Tabel packet_data

FIELD	***TYPE***	***LENGTH***	***CONSTRAINT***
Packet_ID	varchar	50	*Primary Key*
Proccess_Date	varchar	10	
*Source*_IP	varchar	15	
Destnation_IP	varchar	15	
*Source*_Port	bigint	8	
*Destination*_Port	bigint	8	
Packet_Byte	bigint	8	
Protocol_Type	varchar	5	

3.4. Desain Antarmuka Aplikasi

Dalam perancangan aplikasi tidak hanya dibutuhkan perancangan sistem, namun dibutuhkannya perancangan antarmuka yang bertujuan untuk mengetahui apakah antarmuka yang dibuat dapat mempermudah mudah interaksi antara pengguna dengan aplikasi yang akan dibuat, selain itu juga memberikan kemudahan kepada para pembuat aplikasi untuk melakukan penempatan informasi mana yang harus ditampilkan dan informasi mana yang tidak perlu ditampilkan dalam aplikasi identifikasi VoIP ini. Sehingga dalam melakukan rancangan antarmuka ini diharapkan mampu memenuhi aspek-aspek seperti mudah dimengerti dan sederhana, tidak harus melalui prosedur yang terlalu lama, dan memenuhi informasi yang terlibat dalam sistem. Berikut adalah perancangan antarmuka pada halaman utama aplikasi identifikasi lalu-lintas data VoIP Skype.

3.4.1. Desain Antarmuka Sniffer Client

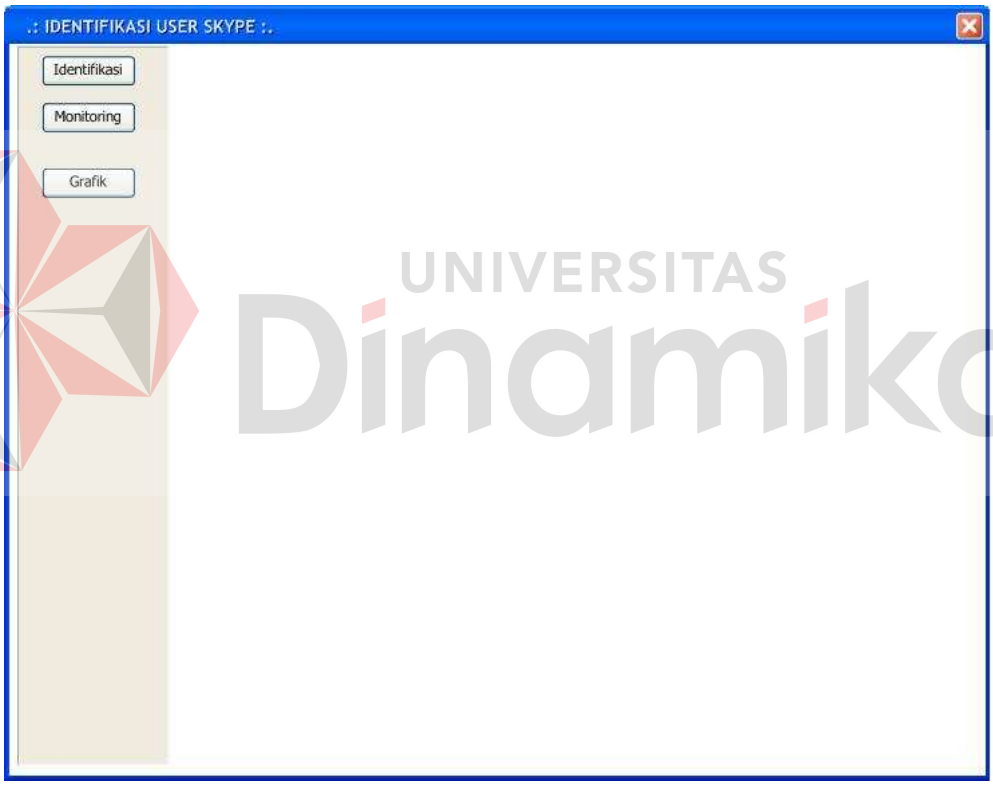
Gambar 3.17 Desain Antarmuka Sniffer Client

Antarmuka Sniffer Client (gambar 3.17) digunakan untuk meng-*capture* semua paket dari *network interface card* pada tiap-tiap komputer. Terdapat dua masukan yang harus diisi yaitu *combobox network interface* yang berisi *network*

interface card komputer yang diamati, dan *time limit* yang digunakan untuk memberi batas waktu kapan proses meng-*capture* tersebut selesai.

Time countdown untuk memberikan infomasi perhitungan mundur waktu untuk mengetahui kapan proses meng-*capture* paket selesai. Sedangkan *the number of data* adalah untuk mengetahui berapa seluruh jumlah paket yang diamati pada *network interface card* komputer yang diamati.

3.4.2. Desain Antarmuka *Form* Utama



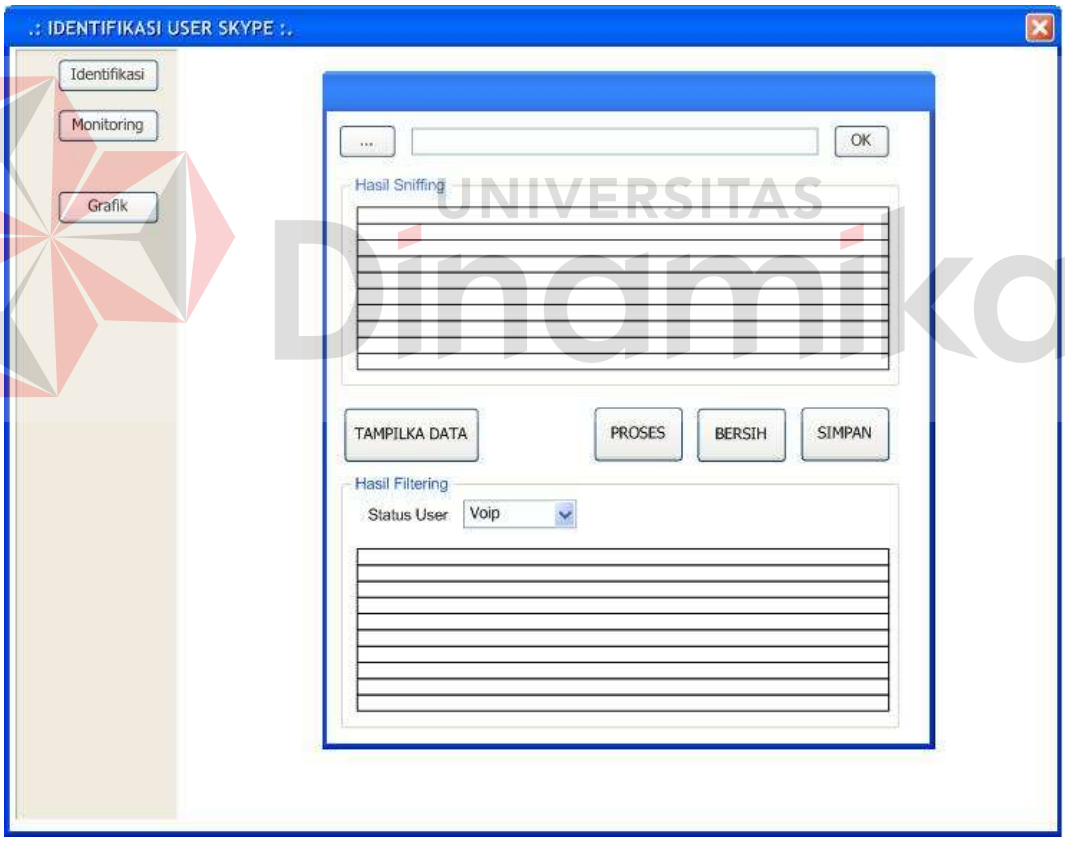
Gambar 3.18 Desain Antarmuka *Form* Utama

Pada antarmuka main program (gambar 3.18) terdapat tiga *link* yang menghubungkan antarmuka utama dengan *form-form* pada antarmuka selanjutnya. Ketiga *link* tersebut adalah Identifikasi, *Monitoring*, dan Grafik. Jika *link* Identifikasi dipilih, maka akan muncul antarmuka indentifikasi *user*. Jika *link Monitoring* dipilih, maka akan muncul antarmuka *monitoring user*. Begitu juga

dengan *link* untuk Grafik dipilih, maka akan muncul antarmuka *report identification*.

3.4.3. Desain Antarmuka Identifikasi *User*

Antarmuka identifikasi *user* pada gambar 3.19 digunakan untuk memfilter paket data dari hasil proses *sniffing* yang dilakukan pada masing-masing client sebelumnya (menggunakan sniffer client). Terdapat satu masukan yaitu *path* database untuk mengambil database dari komputer client yang akan disaring paket datanya.



Gambar 3.19 Desain Antarmuka Identifikasi *User*

Setelah memilih database dari client melalui tombol *path* database, selanjutnya adalah tekan tombol OK untuk membuka koneksi ke database. Kemudian tombol TAMPIL DATA ditekan untuk me*load* data dan menampilkan

seluruh data pada database client di DataGridView. Sehingga pengguna mengetahui isi dari paket yang terekam selama proses *sniffing*.

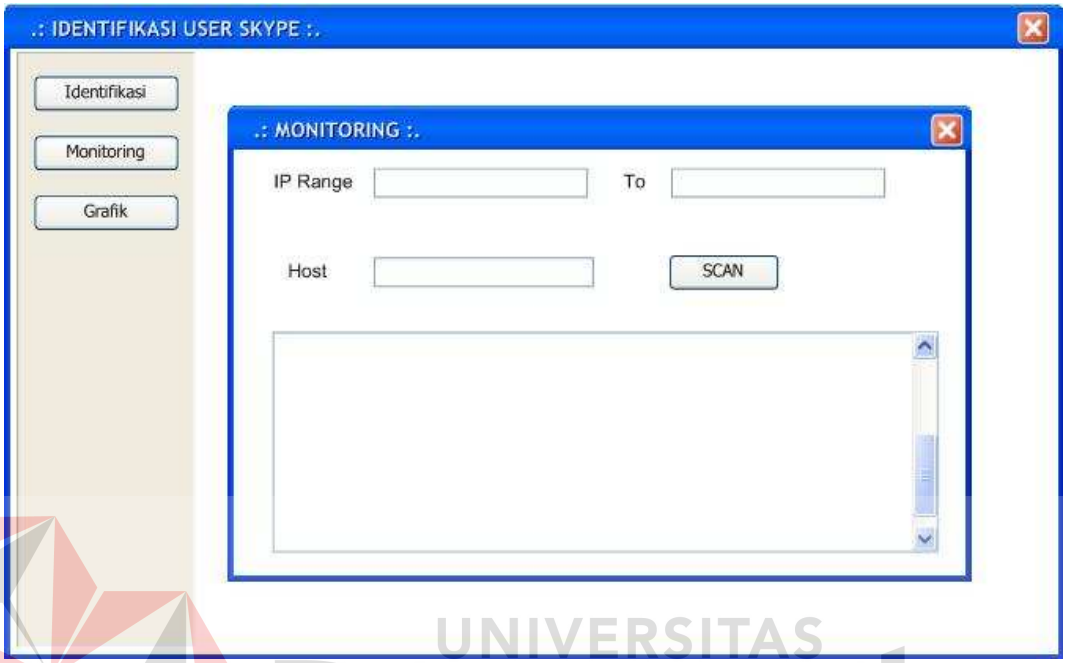
Tombol PROSES berfungsi untuk merubah status combobox yang pada awalnya enablenya bernilai *false* menjadi *true*. Sedangkan combobox status berfungsi untuk melakukan filterisasi data berdasarkan panduan literatur yang ada untuk mengetahui paket Skype. Combobox terdiri dari dua pilihan yaitu login dan VoIP. Setelah data terfilter, data tersebut akan muncul pada data gridview yang berada dibawah combobox. Setelah proses filter selesai data hasil filter tersebut disimpan dengan cara menekan tombol SIMPAN. Seluruh paket data yang didapat dari proses *sniffing* yang berhasil difilter oleh antarmuka ini, disimpan pada basis data yang terdapat pada komputer server.

Jika data sudah disimpan, untuk mengambil data dari komputer client yang lain maka tombol BERSIH harus ditekan untuk membersihkan semua isian pada antarmuka ini.

3.4.4. Desain Antarmuka *Monitoring User*

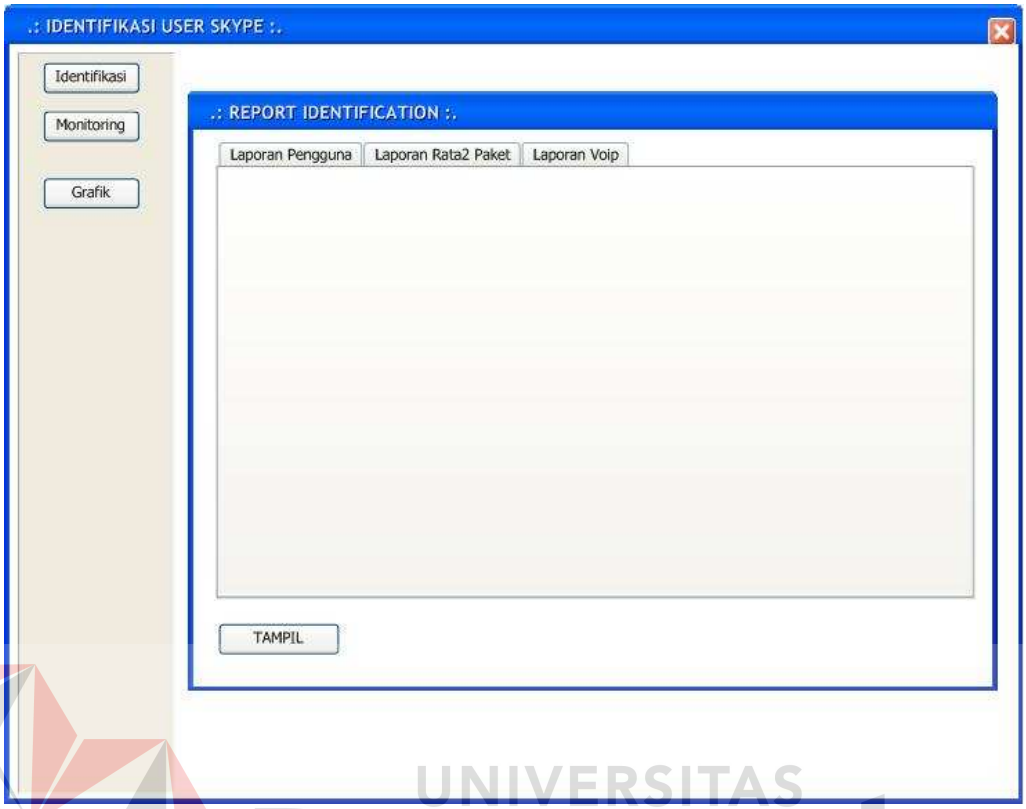
Antarmuka *monitoring user* (gambar 3.20) digunakan untuk melihat alamat IP, komputer mana yang hidup dan sedang menggunakan Skype serta yang tidak. Sehingga dapat diketahui *user* mana yang sedang menggunakan fasilitas VoIP Skype dan yang tidak. Terdapat dua masukan yang harus diisi oleh *user*, yaitu *range* IP untuk menentukan batasan alamat IP awal dan akhir. Range IP diatur *default* dan dapat diubah apabila *user* ingin merubah *Range* IP yang hendak discan. Sehingga user dapat menentukan sendiri *range* ip yang akan dis*can*. Tombol *SCAN* digunakan untuk memulai *scan* ip pada jaringan dan hasil scan ditampilkan pada ListView. Sehingga dapat diketahui jumlah komputer yang

sedang menggunakan fasilitas VoIP pada aplikasi Skype. Dalam ListView akan terlihat nama komputer beserta nomor IP dan status *user* baik yang sedang *login* atau yang menggunakan VoIP.



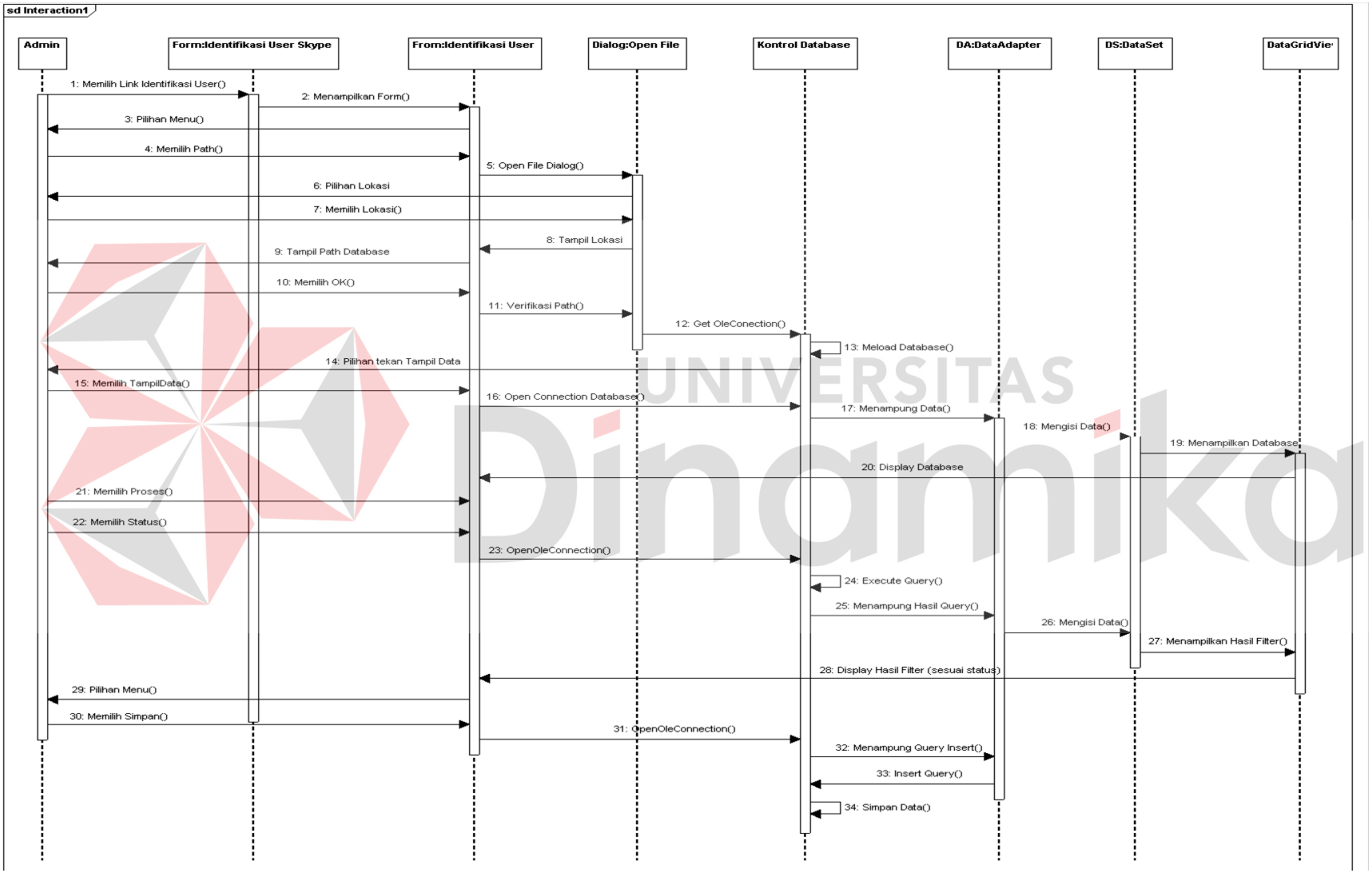
Gambar 3.20 Desain Antarmuka *Monitoring User*

3.4.5. Desain Antarmuka *Report Identification*

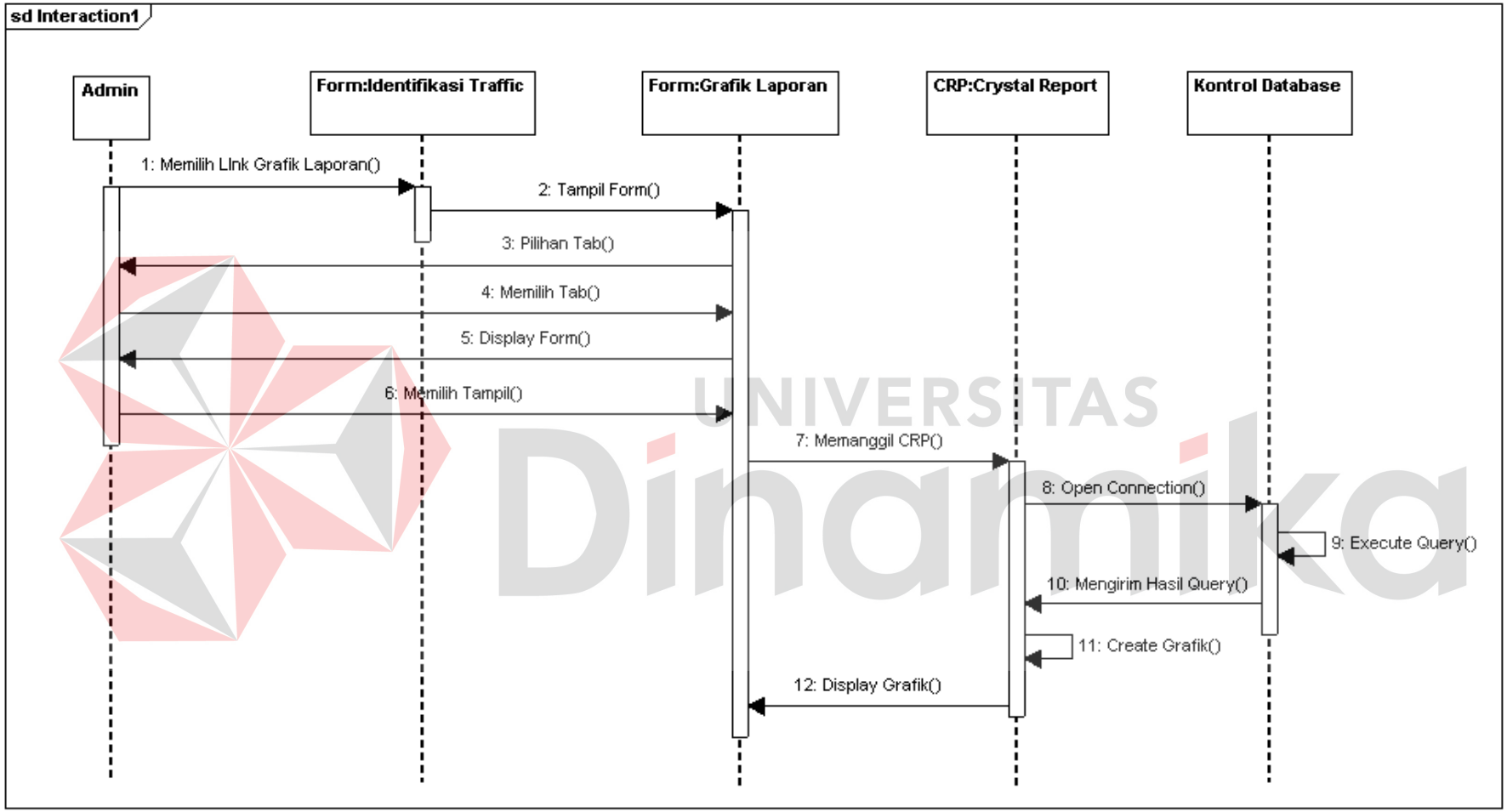


Gambar 3.21 Desain Antarmuka *Report Identification*

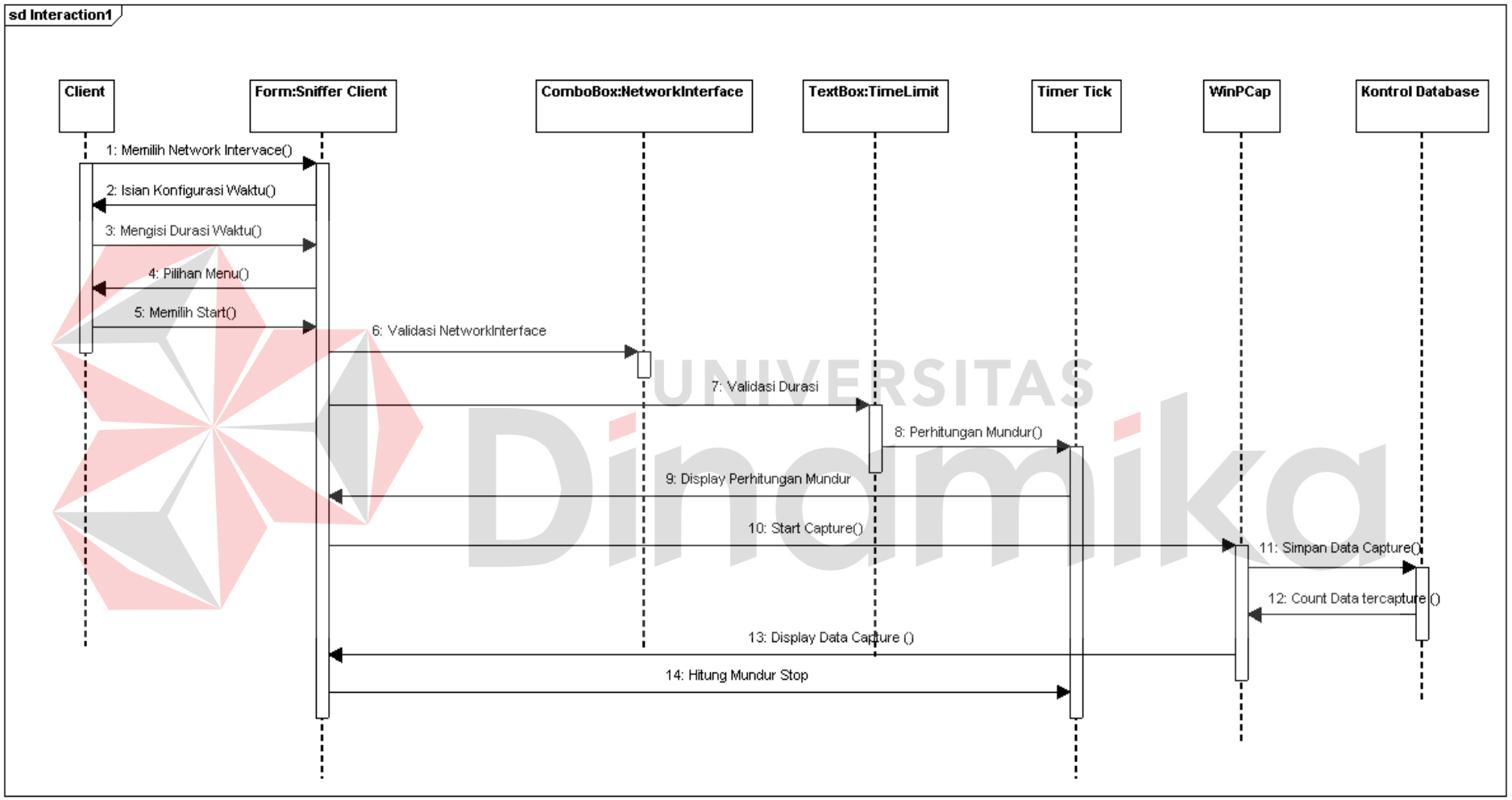
Antarmuka *report identification* (gambar 3.21) adalah desain antarmuka terkahir yang ditampilkan pada aplikasi ini dan fungsinya adalah untuk menampilkan laporan hasil uji coba paket Skype sehingga muncul beberapa diagram laporan. Pada antarmuka ini tersedia tiga pilihan tab laporan. Yang pertama adalah laporan pengguna. Tab yang kedua adalah laporan rata-rata paket, kemudian yang terakhir adalah laporan VoIP.



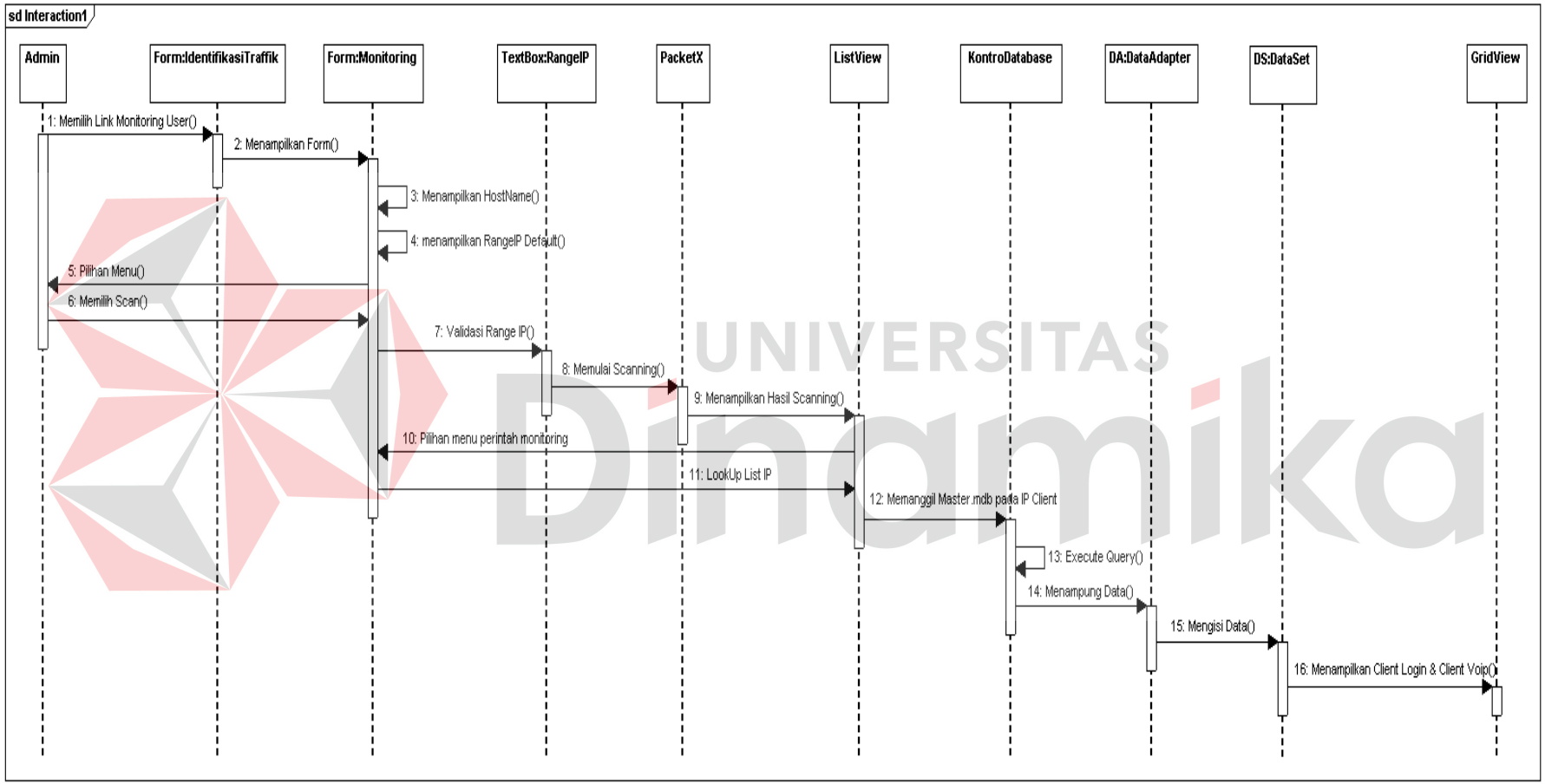
Gambar 3.6 Sequence Diagram Identifikasi User



Gambar 3.11 Sequence Diagram Laporan User Skype



Gambar 3.12 Sequence Diagram Sniffer Client



Gambar 3.8 *Sequence* Diagram *Monitoring* Skype

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Kebutuhan Sistem

Kebutuhan sistem untuk melakukan implementasi piranti lunak identifikasi lalu-lintas data VoIP Skype ini terdapat dua perangkat pendukung, yaitu: piranti lunak dan piranti keras. Berikut adalah daftar kebutuhan piranti lunak dan piranti keras yang dibutuhkan.

4.1.1 Persiapan Piranti Keras

Untuk aplikasi identifikasi lalu-lintas data VoIP Skype ini, terdapat dua kebutuhan akan piranti keras yang digunakan sebagai penunjang aplikasi. Kebutuhan yang pertama adalah piranti keras untuk komputer client, dan yang kedua adalah piranti keras untuk komputer server atau komputer yang melakukan proses identifikasi lalu-lintas data VoIP Skype. Sedangkan yang terakhir adalah kebutuhan piranti keras jaringan.

Kebutuhan piranti keras yang digunakan oleh komputer client adalah sebagai berikut:

a. Processor Pentium 4 2.4 Ghz.

b. Harddisk 60 GB.

c. Memori 512 MB.

d. VGA Card 32 MB.

e. LAN Card.

Sedangkan kebutuhan piranti keras yang digunakan untuk komputer server adalah sebagai berikut:

a. Processor Dual Core 3.0 Ghz.

b. Memori 512 MB (rekomendasi 1 GB).

c. VGA Card 64 MB.

f. Harddisk 80 GB.

g. Monitor SVGA dengan resolusi 1024 x 800.

h. LAN Card.

Kebutuhan yang terakhir adalah kebutuhan piranti keras jaringan yang digunakan yaitu sebagai berikut:

a. Switch dengan 8 port.

b. ADSL modem atau Access Point.

c. Kabel RJ-45 (panjang sesuai dengan kebutuhan).

4.1.2 Persiapan Piranti Lunak

Untuk menjalankan aplikasi identifikasi lalu-lintas data VoIP Skype sesuai dengan yang diharapkan, komputer client dan server memerlukan kebutuhan perangkat lunak sebagai berikut :

a. Sistem operasi Microsoft Windows XP Home atau Profesional.

b. Microsoft.NET Framework 2.0.

c. Microsoft Access 2003

d. WinPcap 4.0.2 (khusus komputer client)

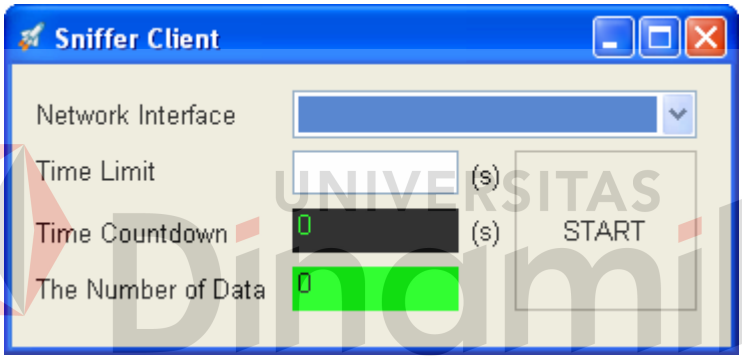
e. Visual Basic.Net 2008

f. Aplikasi Skype

4.2 Implementasi

Setelah selesai melakukan konfigurasi baik piranti keras maupun piranti lunak, langkah selanjutnya adalah melakukan proses evaluasi pada aplikasi identifikasi lalu-lintas data VoIP Skype yang telah dibangun. Agar proses implementasi aplikasi menjadi lebih mudah dan terorganisir, maka proses implementasi dikelompokkan berdasarkan form atau aktifitas yang dimiliki oleh aplikasi ini.

4.2.1 Form Sniffer Client



Gambar 4.1 Form Sniffer Client

Form ini digunakan untuk melakukan *sniffing* proses pada tiap-tiap komputer di dalam suatu jaringan. Terdapat dua masukan yaitu *network interface card* yang ingin diamati dan waktu pengamatan. Setelah *network interface card* dipilih dan waktu pengamatan diisi, tombol *start* ditekan untuk memulai proses *capturing data*.

Apabila tombol *start* ditekan, form ini akan otomatis tersembunyi atau *hidden* karena bertujuan agar tidak mengganggu pengguna komputer selama *sniffing* proses berjalan. Tetapi dapat dimunculkan kembali pada *system tray* yang terletak pada pojok kanan bawah tampilan *operating systems*, atau lebih dikenal

dengan *notify icon*. Klik kanan *notify icon* tersebut, maka akan muncul dua pilihan yaitu *show* atau *hidden*. *Show* berfungsi untuk menampilkan kembali form ini dan *hidden* untuk meneyembunyikannya.

Aplikasi sniffer client ini, akan selesai dengan otomatis sesuai dengan masukan waktu pengamatan tadi. Sehingga benar-benar tidak mengganggu aktivitas pengguna komputer sama sekali. Aplikasi ini terpisah dengan aplikasi Identifikasi *User* Skype pada komputer server, tetapi menjadi satu bagian dari aplikasi tugas akhir ini.

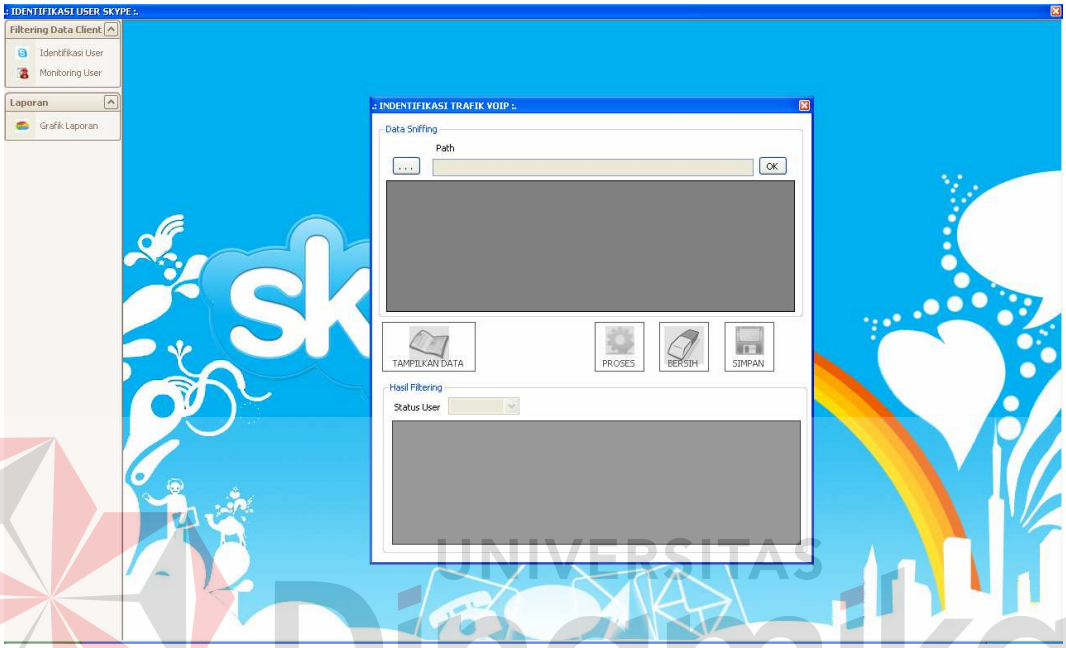
4.2.2 Form Utama

Gambar 4.2 Identifikasi *User* Skype

Pada saat pertama kali program dijalankan, form pertama yang akan muncul adalah form Identifikasi *User* Skype (gambar 4.2). Pada pojok kiri atas dari form identifikasi *user* Skype ini terdapat tiga pilihan *link* yaitu Identifikasi *User*, *Monitoring User* dan Grafik Laporan. Jika pilihan Identifikasi *User* dipilih, maka akan muncul form Identifikasi *User*. Begitu juga dengan pilihan *Monitoring*

User dipilih, maka akan muncul form *Monitoring*. Jika pilihan Grafik Laporan dipilih, maka akan muncul form *Report Identification* untuk data pengguna Skype

4.2.3 Form Identifikasi *User*



Gambar 4.3 Identifikasi *User*

Setelah melakukan pilihan Identifikasi *User*, maka akan muncul form Identifikasi *User* seperti gambar di atas (gambar 4.3). Form ini adalah form yang akan memfilter paket data dari hasil proses *sniffing* yang dilakukan pada masing-masing client sebelumnya (menggunakan sniffer client). Untuk mengambil database dari komputer client yang akan disaring paket datanya, caranya ialah dengan memasukkan *path database* atau letak dari database lokal itu berada..

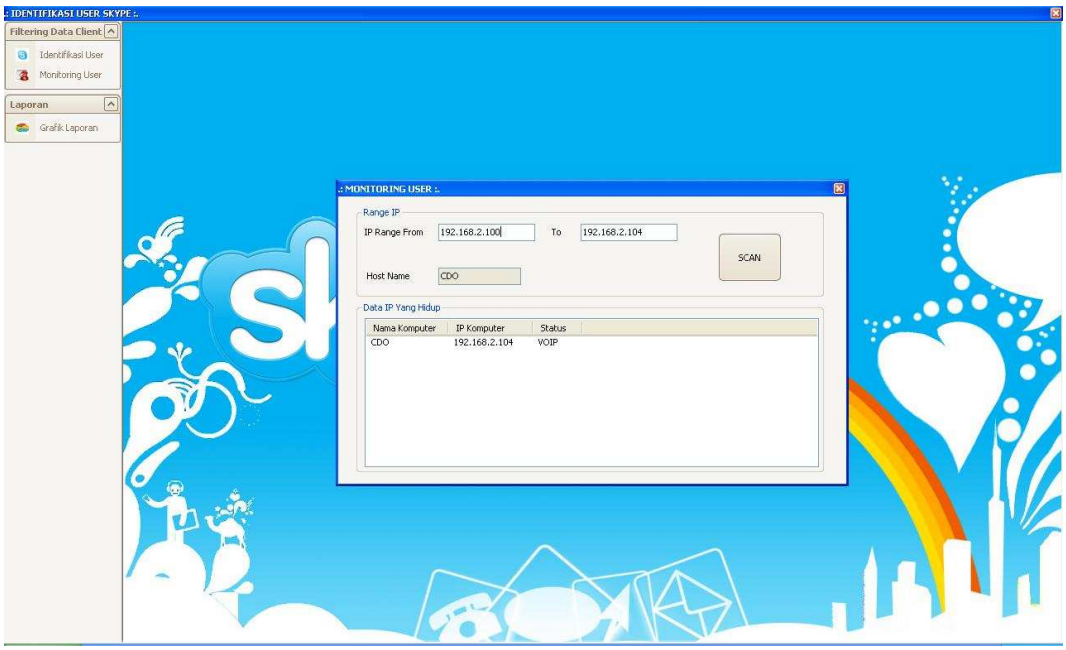
Setelah memilih letak database dari client lalu menekan tombol OK untuk merubah koneksi ke database lokal tersebut. Dengan menekan pilihan tombol TAMPIL DATA, maka form me*load* data dan seluruh data pada *database client*

akan ditampilkan di *datagridview* pada form ini. Sehingga pengguna dapat mengetahui seluruh isi paket yang terekam selama proses *sniffing*.

Kemudian selanjutnya ialah melakukan pilihan PROSES berfungsi untuk merubah status combobox yang pada awalnya *enable*nya bernilai false menjadi true, sehingga dialog combobox menjadi aktif. Combobox status ini berfungsi untuk melakukan filterisasi data berdasarkan panduan literatur yang ada untuk mengetahui paket Skype. Combobox terdiri dari dua pilihan yaitu login dan VoIP. Pengguna memilih salah satu dari kedua status tersebut, kemudian data akan terfilter. Setelah data terfilter, data yang berisi paket-paket Skype tersebut akan secara otomatis muncul pada data gridview yang berada dibawah combobox. Setelah proses filter selesai data hasil filter tersebut disimpan dengan cara melakukan pilihan pada tombol SIMPAN. Seluruh paket data yang didapat dari proses snifing yang berhasil difilter oleh form ini disimpan pada basis data yang terdapat pada komputer server.

Jika data sudah disimpan, untuk mengambil data dari komputer client yang lain maka tombol BERSIH harus ditekan untuk membersihkan semua isian pada form ini.

4.2.4 Form Monitoring *User*

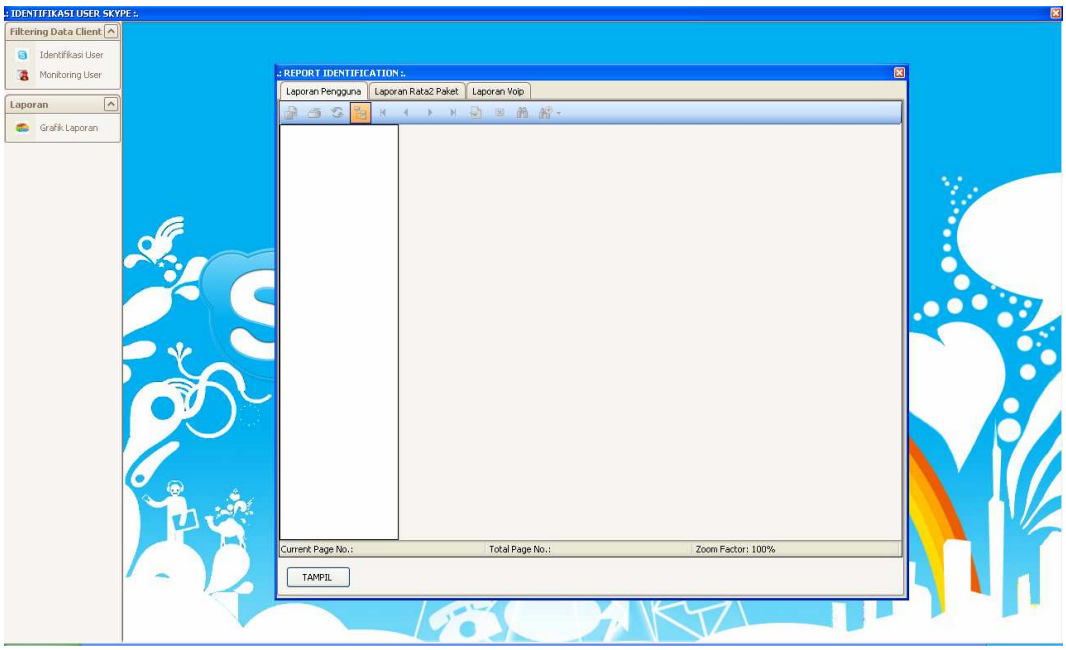


Gambar 4.4 Monitoring *User*

Form *monitoring user* (gambar 4.4) akan muncul ketika *link Monitoring* pada form utama dipilih. Form ini digunakan untuk melihat alamat IP, komputer mana yang hidup dan sedang menggunakan Skype serta yang tidak. Sehingga dapat diketahui *user* mana yang sedang menggunakan fasilitas VoIP Skype dan yang tidak. Terdapat dua masukan yang harus diisi oleh *user*, yaitu range IP untuk menentukan batasan alamat IP awal dan akhir.

Pada form monitoring, tombol *SCAN* digunakan untuk memulai *scan* IP pada jaringan. Pada saat proses *scan* ListView akan menampilkan nama komputer beserta alamat IP dan status komputer tersebut. Sehingga dapat diketahui jumlah komputer yang sedang menggunakan fasilitas VoIP serta status *user*. Terdapat dua status yang akan terlihat pada, yaitu login dan VoIP.

4.2.5 Form *Report* Identification



Gambar 4.5 *Report* Identification

Form *report identification* (gambar 4.5) di atas adalah form terkahir yang ditampilkan pada aplikasi ini melalui *link* Grafik Laporan yang ada pada form utama. Fungsi dari form ini adalah untuk menampilkan laporan hasil uji coba paket Skype sehingga muncul beberapa diagram laporan. Pada form ini tersedia tiga pilihan tab laporan. Yang pertama adalah laporan pengguna yang melaporkan jumlah *user* Skype per tanggal. Tab yang kedua adalah laporan rata-rata paket yang melaporkan rata-rata paket byte dari Skype di setiap *user*, kemudian yang terakhir adalah laporan VoIP. Laporan VoIP ini berisi laporan pengguna VoIP per IP. Untuk menampilkan pada setiap tab, *user* menekan tombol TAMPIL sehingga diagram akan muncul.

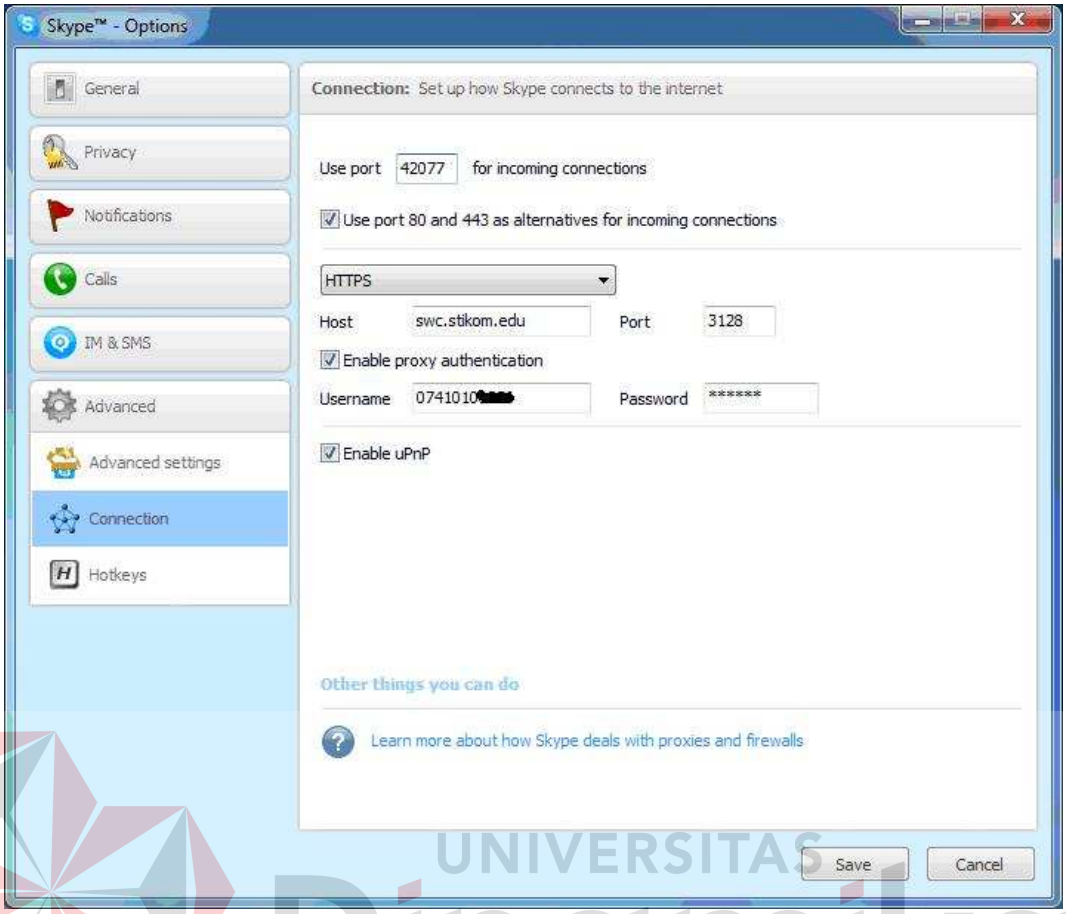
4.3 Evaluasi

Evaluasi aplikasi ini dilakukan dengan menggunakan dua titik pengamatan yaitu pengamatan penggunaan aktivitas Skype pada masing-masing komputer *client* dan akses internet ke Skype. Desain jaringan menggunakan satu komputer server dan lima komputer *client*. Dari hasil evaluasi aplikasi ini dapat mengetahui hasil yang menjawab semua masalah pada analisis sistem pada bab 3.

4.3.1 Capturing Data Pada Client

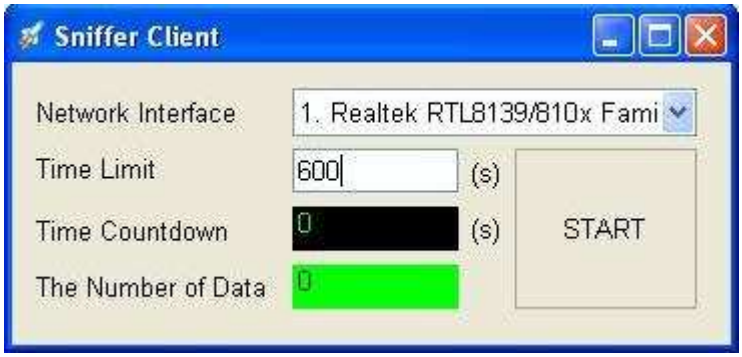
Pada sisi client, hal yang pertama dilakukan adalah melakukan proses *sniffing* untuk meng*capture* seluruh paket data. Untuk melaksanakan hal tersebut, penulis mensimulasikan sebuah jaringan LAN yang telah terkoneksi internet. Dalam jaringan, terdapat satu komputer server dan lima komputer client. Masing-masing komputer client telah terinstall dua program, yaitu aplikasi Skype dan Sniffer Client.

Untuk memulai proses *capturing*, persiapan pertama yaitu aplikasi Skype dalam posisi *ready* atau siap dijalankan. Sebelumnya pada aplikasi Skype klik *Tool* kemudian pilih *Option*. Sehingga akan muncul kotak dialog *Skype Option* seperti pada gambar 4.6. Pada *Navigation Bar* di posisi kiri, pilih *Advanced* sehingga muncul tiga menu yaitu *Advance Setting*, *Connection* dan *Hotkeys*. Pilih menu *Connection* sehingga muncul dialog *Connection* pada sisi kanan kotak dialog. Untuk Use Port, diisi 42077 sebagai port untuk *incoming connections*. Kemudian mencentang "*Use port 80 and 443 as alternatives for incoming connections*" karena port 80 dan 443 ini digunakan sebagai parameter uji coba. Untuk menyimpan *setting* ini tekan tombol *Save*.



Gambar 4.6 *Setting Connection* Aplikasi Skype

Persiapan kedua adalah aplikasi Sniffer Client diaktifkan pada masing-masing komputer. Kemudian konfigurasi *network interface* dan *time limit* pada seluruh client diatur sama. Untuk *network interface* dipilih berdasarkan konfigurasi yang ada pada masing-masing client, kemudian untuk *time limit*nya diatur 600 s (600 sekon atau kurang lebih 10 menit). Kemudian tombol *start* ditekan, sehingga proses *capturing* mulai bergulir. Setelah start dimulai, masing-masing client diminta untuk *login account* Skypenya masing-masing. Proses login Skype ini diperlukan untuk mengidentifikasi paket Skype.



Gambar 4.7 Sniffer Client

Setelah login ke Skype selesai dilaksanakan masing-masing client, selanjutnya client disimulasikan menjadi dua kondisi sebagai berikut:

1. Satu client hanya *stay connected* saja namun tidak melakukan aktivitas apapun.
2. Empat client melakukan aktivitas VoIP.

Setelah waktu 600 s berakhir, Sniffer Client akan menyimpan database secara otomatis dengan format nama default sniffer.mdb, karena untuk mengantisipasi salah memasukkannbasis data lokal yang mempunyai *extensi* sama yaitu .mdb atau Microsoft Access file.

4.3.2 Identifikasi, Monitoring, dan Diagram Laporan pada Server

Setelah melakukan *capturing* data pada client, selanjutnya admin melakukan *download* basis data lokal yang terdapat pada komputer client melalui jaringan. Seluruh basis data lokal yang di*download* dikumpulkan pada komputer server dan dipisah pada folder-folder yang sesuai dengan nama komputer client. Hal ini dilakukan karena semua file database memiliki nama yang sama, yaitu sniffer.mdb.

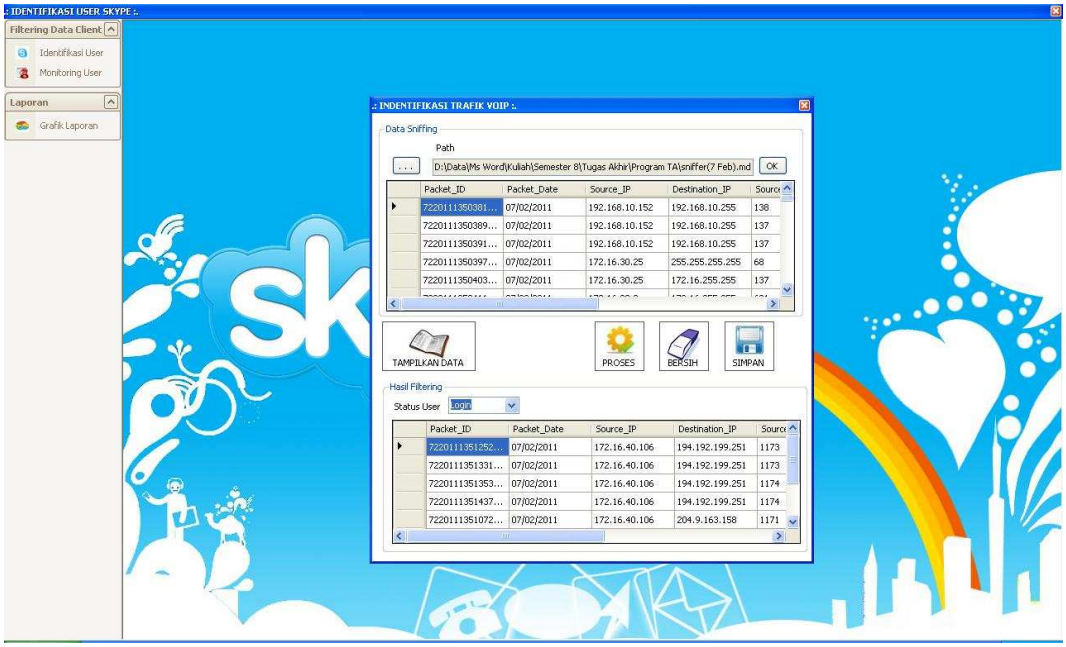
Untuk memulai identifikasi pada form utama dipilih link identifikasi *user*. Pada form identifikasi *traffic* VoIP, tekan tombol path  dan pilih database lokal sesuai dengan komputer yang diamati. Kemudian klik OK untuk dapat mengakses database, dan menekan tombol TAMPILKAN DATA untuk menampilkan database ke DataGridView. Selanjutnya menekan tombol PROSES dan memilih status combobox (login atau VoIP). Database akan terfilter dan ditampilkan langsung pada DataGridView yang di bawah. Potongan SQL command untuk combobox login adalah sebagai berikut :

```
SELECT   *
FROM     packet_data
WHERE    destination_port in (80,443) AND destination_ip
         in('194.192.199.251','194.165.188.101','212.187.172
         .78','204.9.163.158')
```

Sedangkan potongan SQL command untuk combobox VoIP adalah sebagai berikut :

```
SELECT   *
FROM     packet_data
WHERE    packet_byte between 80 and 640 AND source_port >
         1024 AND protocol_type = 'tcp'
```

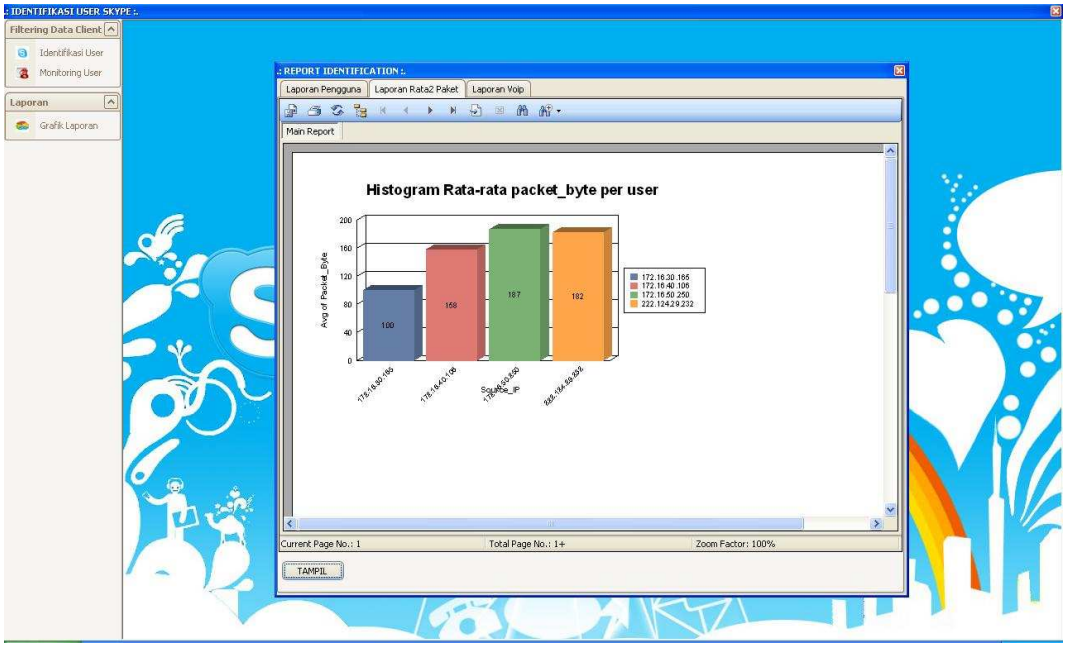
Untuk menyimpan data hasil filter, admin menekan tombol SIMPAN. Pemfilteran data ini dilakukan untuk seluruh data *sniffing* (yang tersimpan dalam folder-folder pada komputer server). Untuk memulai proses *filtering* dari awal, admin menekan tombol BERSIH untuk membersihkan semua isian pada form ini.



Gambar 4.8 Form Identifikasi Traffic VoIP

Setelah semua proses filtering selesai, admin mengakses form Grafik Laporan. Di dalam form *report identification* terdapat tiga tab diagram laporan.

1. **Tab Laporan Pengguna**, klik tombol TAMPIL untuk menampilkan diagram jumlah *user* Skype per tanggal
2. **Tab Laporan Rata-Rata Paket**, klik tombol TAMPIL untuk menampilkan diagram rata-rata paket byte dari Skype di setiap *user*,
3. **Tab Laporan VoIP**, klik tombol TAMPIL untuk menampilkan diagram laporan pengguna VoIP per IP.

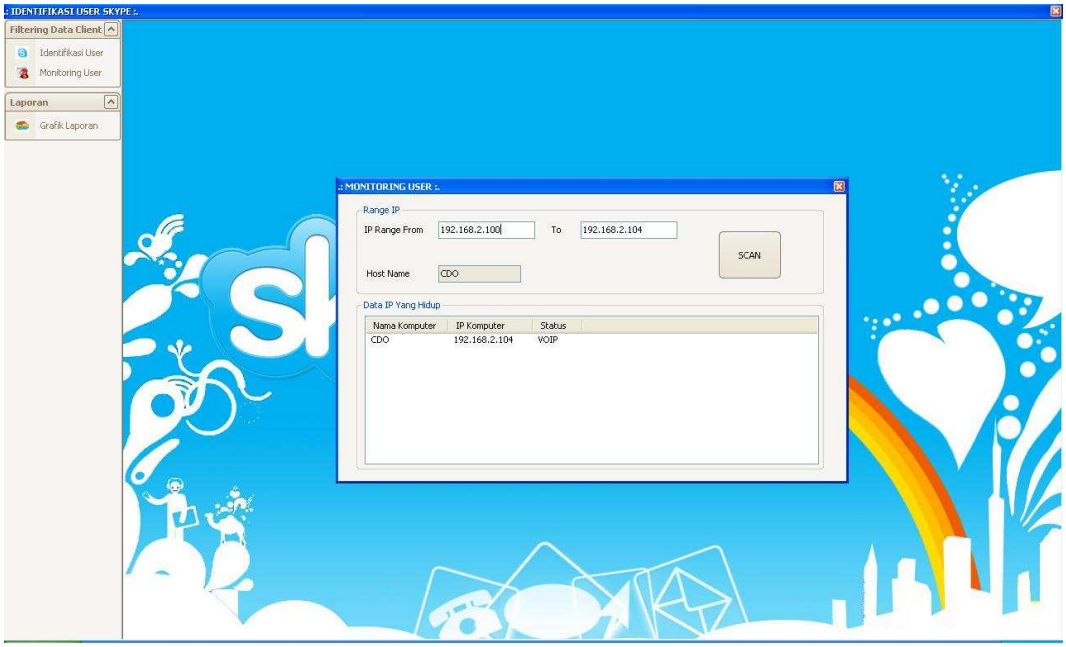


Gambar 4.9 Form *Report Identification*

Hasil dari ketiga tab laporan tersebut akan dijelaskan lebih lanjut pada analisa hasil evaluasi.

Untuk form monitoring *user* hanya digunakan untuk men*scan* IP komputer client yang sedang menyala. Pertama-tama admin dapat merubah besaran range IP pada *textbox* untuk menentukan range IP yang akan di*scan*. Selanjutnya menekan *SCAN*, sehingga proses *monitoring* bergulir. Dari sinilah didapatkan IP komputer client yang sedang menyala seperti pada gambar 4.10.

Untuk mengetahui client mana yang sedang menggunakan Skype dan yang tidak, tampak pada kolom status yang ada pada ListView, sehingga admin dapat mengetahui nama komputer beserta alamat IP *user* yang sedang menggunkan fasilitas VoIP pada aplikasi Skype.



Gambar 4.10 Form *Monitoring user*

4.3.3 Analisa Hasil Evaluasi

Dari hasil evaluasi sistem identifikasi *traffic* VoIP Skype ini ditemukan beberapa perbandingan yang mendasar baik pada saat login maupun saat VoIP. Dari data yang dihasilkan melalui proses identifikasi untuk pengguna Skype dapat diketahui berapakah besaran paket data dari Skype. Paket data tersebut antara lain berupa paket untuk login dan paket untuk melakukan komunikasi VoIP. Berdasarkan uji coba yang menitik beratkan pada port 80 dan 443 sebagai titik pengamatan untuk mengetahui paket data Skype yang menuju server Skype, dapat diketahui informasi tentang data apa saja yang dibawa oleh Skype. Berikut analisa yang dilaksanakan

1. Hasil Analisa Skype Traffic pada Proses Login

Pada saat melaksanakan proses login, seorang client Skype pasti mengkoneksikan dirinya dengan webserver dari Skype, yaitu di ui.Skype.com. Dalam literatur Dongyan Zhang dkk., dia mendefinisikan IP address dari webserver ui.Skype.com adalah sebagai berikut :

a. 193.88.6.228
b. 212.187.172.228
c. 212.72.49.136
d. 217.159.236.228

Namun ketika pengujian IP address webserver ini menggunakan *command prompt* dan melalui website **www.hcidata.info/host2ip.cgi,** ternyata IP address server telah berubah. IP address dari webserver Skype menjadi seperti berikut ini :

a. 204.9.163.158
b. 212.187.172.78
c. 194.192.199.251
d. 194.165.188.101

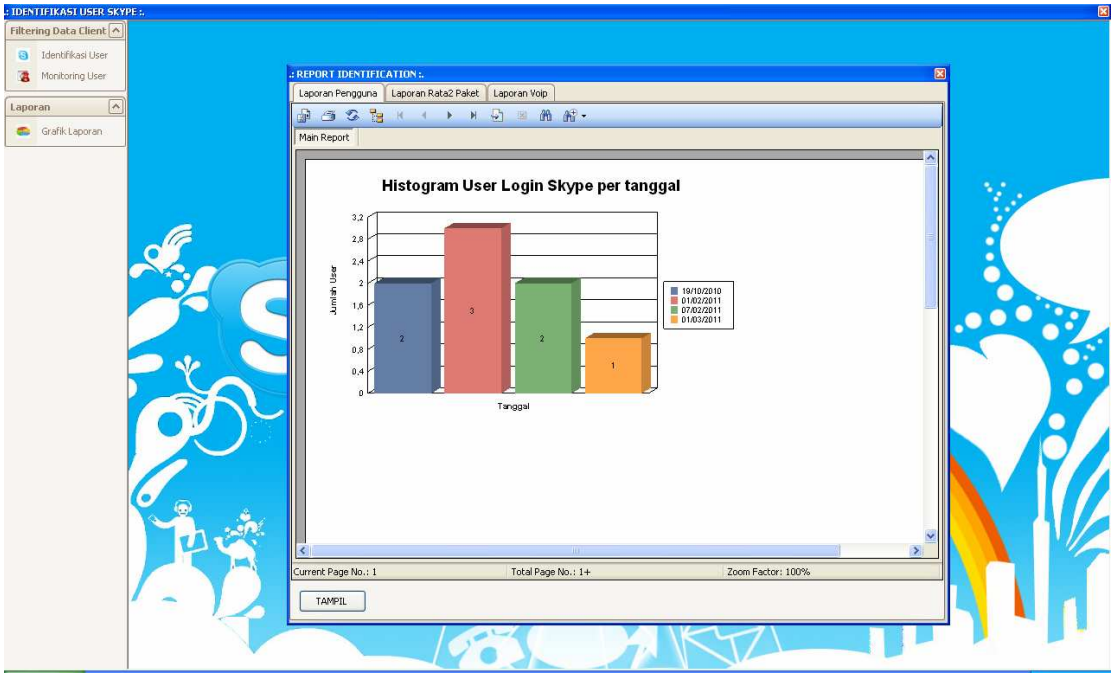
Keempat IP address terbaru inilah yang dijadikan sebagai parameter IP destination untuk mengidentifikasi paket Skype dalam form identifikasi *user*.

Di literatur yang sama, Dongyan menyatakan bahwa saat login TCP connection adalah port 80 untuk akses HTTP dan port 443 untuk akses HTTPS. Kedua port ini yang diujikan sebagai parameter untuk mengetahui pola aplikasi Skype saat berkomunikasi dengan server untuk melakukan autentifikasi *user*.

Packet_ID	Packet_Date	Source_IP	Destination_IP	Source_Port	Destination_Po	Packet_Byte	Protocol_Type
1220111435502	01/02/2011	222.124.29.232	172.16.40.106	3128	1938	99	tcp
1220111435516	01/02/2011	172.16.40.106	222.124.29.232	1939	3128	124	tcp
1220111435517	01/02/2011	172.16.40.106	222.124.29.232	1939	3128	479	tcp
1220111435522	01/02/2011	222.124.29.232	172.16.40.106	3128	1939	315	tcp
1220111435534	01/02/2011	172.16.40.106	222.124.29.232	1940	3128	92	tcp
1220111435538	01/02/2011	222.124.29.232	172.16.40.106	3128	1940	99	tcp
1220111435544	01/02/2011	172.16.40.106	222.124.29.232	1942	3128	230	tcp
1220111435547	01/02/2011	172.16.40.106	204.9.163.158	1943	80	28	tcp
1220111435552	01/02/2011	172.16.40.106	204.9.163.158	1943	80	28	tcp
1220111435557	01/02/2011	172.16.40.106	204.9.163.158	1943	80	28	tcp
1220111435572	01/02/2011	172.16.40.106	194.192.199.25	1946	443	28	tcp
1220111435577	01/02/2011	172.16.40.106	194.192.199.25	1946	443	28	tcp
1220111435582	01/02/2011	172.16.40.106	194.192.199.25	1947	80	28	tcp
1220111435587	01/02/2011	172.16.40.106	194.192.199.25	1947	80	28	tcp
1220111445455	01/02/2011	172.16.40.106	172.16.50.250	1981	23480	518	tcp
1220111445462	01/02/2011	172.16.50.250	172.16.40.106	23480	1981	187	tcp
1220111445462	01/02/2011	172.16.40.106	172.16.50.250	1981	23480	90	tcp
1220111445508	01/02/2011	172.16.40.106	222.124.29.232	1938	3128	483	tcp
1220111446508	01/02/2011	172.16.40.106	222.124.29.232	1938	3128	483	tcp
1220111448508	01/02/2011	172.16.40.106	222.124.29.232	1938	3128	483	tcp
1910201013450	19/10/2010	222.124.29.232	172.16.30.165	3128	1207	315	tcp
1910201013451	19/10/2010	222.124.29.232	172.16.30.165	3128	1208	99	tcp
1910201013451	19/10/2010	222.124.29.232	172.16.30.165	3128	1208	101	tcp
1910201013452	19/10/2010	222.124.29.232	172.16.30.165	3128	1208	294	tcp
1910201013452	19/10/2010	172.16.30.165	222.124.29.232	1208	3128	143	tcp
1910201013453	19/10/2010	172.16.30.165	212.187.172.78	1212	443	28	tcp

Gambar 4.11 Capture data hasil filter “login”

Berdasarkan gambar 4.11 diketahui bahwa pada saat Login masing-masing client mengakses port tujuan 80 atau 443 (*destination port*) yang menuju webserver Skype yaitu IP destination dari Skype. Dari data tersebut juga diketahui jumlah paket_byte yang dikirim dan diterima oleh masing-masing client beserta dengan jenis protokol_type. Dari hasil filter pada kotak *highlight* merah gambar 4.10, diketahui bahwa untuk melakukan proses login, seorang client Skype membawa besaran paket byte yang relatif kecil yaitu sebesar 28 Byte. Besarnya paket byte saat uji coba login ini sesuai dengan literatur Dongyan Zhang dkk. Yang menyatakan besarnya paket byte saat login adalah antara 25 byte sampai dengan 39 byte.



Gambar 4.12 Laporan *User* Skype per Tanggal

Pada gambar 4.12 diagram *user* Skype per tanggal, dapat diketahui ada berapa *user* Skype yang sedang *online* atau telah login Skype. Sehingga admin dapat mengetahui ada berapa client yang *online* Skype pada setiap tanggalnya.

2. Hasil Analisa Skype Traffic VoIP

Parameter untuk mengidentifikasi VoIP pada Skype ini ada dua yaitu *source port* dan *packet byte*. Kedua parameter ini juga disesuaikan dengan literatur. Pada literatur Marcell Parenyi, *port* bebas yang digunakan Skype untuk komunikasi VoIP adalah lebih dari 1024. Hal ini terbukti benar karena pada saat melakukan uji coba VoIP, *Source Port* yang digunakan memang selalu lebih dari 1024. Hasil ini dapat diamati pada gambar 3.13. Sedangkan hasil filtering data *Packet_Byte* VoIP berkisar antara 80 Byte hingga 640 Byte. Hal ini sama dengan yang dituliskan Dongyan pada literaturnya. Kisaran besar paket untuk pengujian

VoIP ini dapat dilihat pada kotak *highlight* merah pada gambar 4.13 database hasil filter uji coba VoIP.

Packet_ID	Packet_Date	Source_IP	Destination_IP	Source_Port	Destination_Po	Packet_Byte	Protocol_Type
1220111435502	01/02/2011	222.124.29.232	172.16.40.106	3128	1938	99	tcp
1220111435518	01/02/2011	172.16.40.106	222.124.29.232	1939	3128	124	tcp
1220111435517	01/02/2011	172.16.40.106	222.124.29.232	1939	3128	479	tcp
1220111435522	01/02/2011	222.124.29.232	172.16.40.106	3128	1939	315	tcp
1220111435534	01/02/2011	172.16.40.106	222.124.29.232	1940	3128	92	tcp
1220111435538	01/02/2011	222.124.29.232	172.16.40.106	3128	1940	99	tcp
1220111435544	01/02/2011	172.16.40.106	222.124.29.232	1942	3128	230	tcp
1220111435547	01/02/2011	172.16.40.106	204.9.163.158	1943	80	28	tcp
1220111435552	01/02/2011	172.16.40.106	204.9.163.158	1943	80	28	tcp
1220111435557	01/02/2011	172.16.40.106	204.9.163.158	1943	80	28	tcp
1220111435572	01/02/2011	172.16.40.106	194.192.199.25	1946	443	28	tcp
1220111435577	01/02/2011	172.16.40.106	194.192.199.25	1946	443	28	tcp
1220111435582	01/02/2011	172.16.40.106	194.192.199.25	1947	80	28	tcp
1220111435587	01/02/2011	172.16.40.106	194.192.199.25	1947	80	28	tcp
1220111445455	01/02/2011	172.16.40.106	172.16.50.250	1981	23480	518	tcp
1220111445462	01/02/2011	172.16.50.250	172.16.40.106	23480	1981	187	tcp
1220111445462	01/02/2011	172.16.40.106	172.16.50.250	1981	23480	90	tcp
1220111445508	01/02/2011	172.16.40.106	222.124.29.232	1938	3128	483	tcp
1220111446508	01/02/2011	172.16.40.106	222.124.29.232	1938	3128	483	tcp
1220111448508	01/02/2011	172.16.40.106	222.124.29.232	1938	3128	483	tcp
1910201013450	19/10/2010	222.124.29.232	172.16.30.165	3128	1207	315	tcp
1910201013451	19/10/2010	222.124.29.232	172.16.30.165	3128	1208	99	tcp
1910201013451	19/10/2010	222.124.29.232	172.16.30.165	3128	1208	101	tcp
1910201013452	19/10/2010	222.124.29.232	172.16.30.165	3128	1208	294	tcp
1910201013452	19/10/2010	172.16.30.165	222.124.29.232	1208	3128	143	tcp
[illegible]							
1910201013454	19/10/2010	172.16.30.165	212.187.172.78	1212	443	28	tcp
1910201013454	19/10/2010	172.16.30.165	212.187.172.78	1212	443	28	tcp
1910201013454	19/10/2010	172.16.30.165	212.187.172.78	1213	80	28	tcp

Gambar 4.13 Database Hasil Filter Uji Coba VoIP

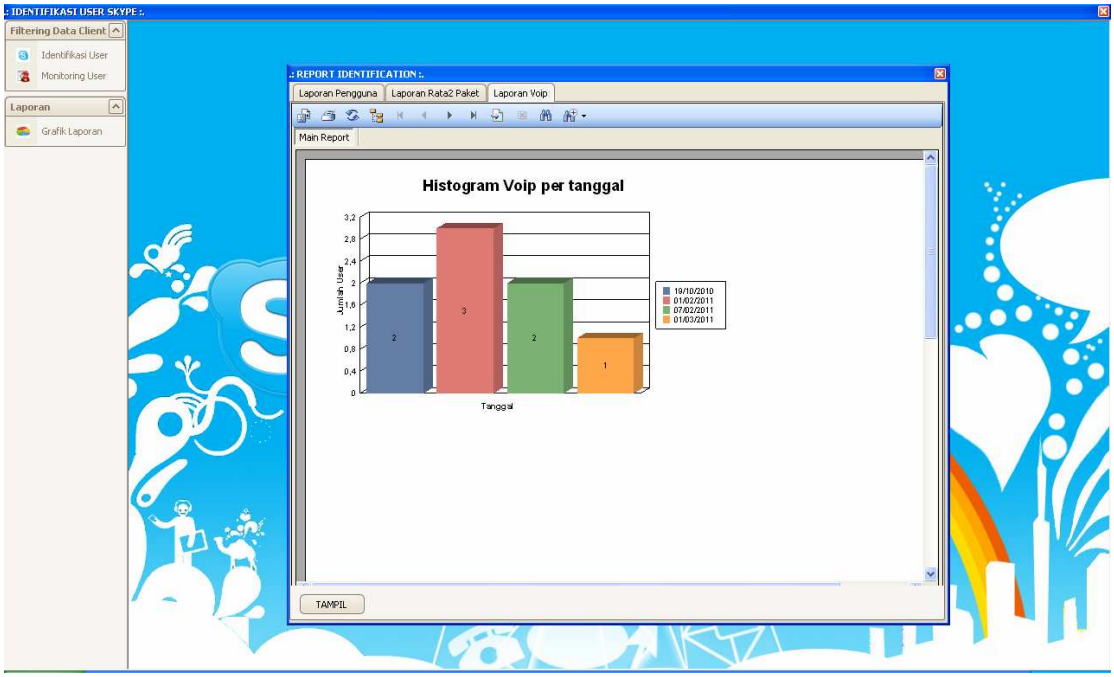
Pada kotak merah (pada panah pengarah), posisi *Packet_Byte* terendah adalah 90 dan yang tertinggi adalah 518. Kisaran *Packet_Byte* ini membukti kebenaran bahwa besar paket byte yang digunakan Skype untuk berkomunikasi VoIP adalah antara 80 dan 640 Byte.

Namun terdapat kekurangan dari aplikasi ini adalah bahwa port proxy yang digunakan user untuk mengakses internet juga ikut tercapture. Karena besarnya port proxy tersebut bernilai lebih dari 1024. Jika di dalam hasil filter terdapat port proxy, maka itu bukan merupakan penanda VoIP pada Skype. Dalam kasus ini, pada data yang tidak valid adalah data yang menunjukkan pada source IP dan destination IP nya terdapat port 3128. Karena port 3128 di sini merupakan proxy untuk akses internet. Sebagai contoh dapat dilihat pada *highlight* bagian

bawah pada tanggal 19/10/2010, terlihat bahwa dari source IP 172.16.30.165 (user) menuju 222.124.29.232 (IP DNS), memiliki source port 1208 menuju port 3128 (port proxy). Data seperti ini yang bukan menunjukkan data komunikasi VoIP.

Data yang menunjukkan komunikasi VoIP adalah seperti pada panah pengarah pada gambar 4.13. Sebagai contoh dapat dilihat pada *highlight* bagian bawah pada tanggal 01/02/2011, terlihat bahwa dari source IP 172.16.40.106 (user) menuju destination IP 17216.50.250 (user), memiliki source port 1961 menuju port 23480. Paket Byte yang dibawa sebesar 518 Byte. Data seperti ini yang menunjukkan data komunikasi VoIP.

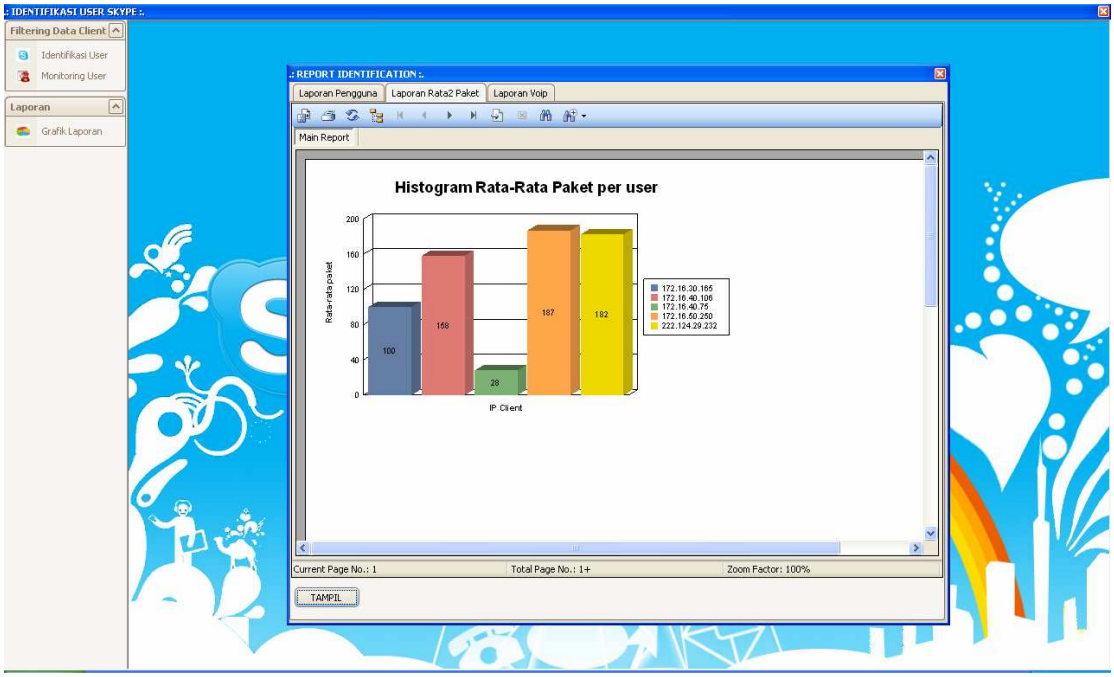
Untuk mengetahui client mana saja yang melakukan VoIP dan yang tidak dapat dilihat pada Grafik Laporan. Laporan *user* VoIP ini terlihat seperti pada gambar 4.14. Pada gambar 4.14 diagram *user* VoIP per tanggal, dapat diketahui ada berapa *user* Skype yang sedang menggunakan komunikasi VoIP. Sehingga pada setiap tanggalnya, admin dapat mengetahui ada berapa client yang sedang melakukan VoIP menggunakan Skype.



Gambar 4.14 Laporan *User* VoIP per Tanggal

3. Hasil Analisa Rata-Rata Paket Skype

Sebagaimana yang telah dijelaskan pada analisa sebelumnya bahwa untuk paket byte saat login cenderung pada posisi 28 Byte, yang mengartikan bahwa kisaran paket byte untuk proses login memang antar 25 Byte sampai 39 Byte. Sedangakan pada saat komunikasi VoIP, kisaran paket byte yang digunakan pada setiap client antara 80 sampai 640 Byte. Sehingga dapat terlihat secara kasar bahwa besarnya paket byte yang dibawa oleh Skype dari proses login hingga melakukan komunikasi VoIP adalah sebesar 25 Byte untuk batas terendah dan 640 Byte untuk batas atasnya. Data ini dapat terlihat jelas pada diagram gambar 3.15.



Gambar 4.15 Laporan Rata-rata Paket_Byte per Tanggal

Pada gambar 4.15 terlihat bahwa di setiap client memiliki rata-rata paket byte yang berbeda-beda tergantung pada banyaknya aktivitas yang dilakukan masing-masing. Misalnya pada clint dengan IP 172.16.50.250 (balok orange) memiliki rata-rata paket tertinggi yakni senilai 187 Byte. Sedangkan untuk client dengan IP 172.16.40.75 (balok hijau) memiliki rata-rata paket terendah yakni senilai 28 Byte. Diagram ini membuktikan bahwa paket Skype tidak pernah lebih rendah dari 25 Byte dan tidak pernah lebih tinggi dari 640 Byte.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah dilakukan uji coba, aplikasi identifikasi lalu-lintas data Skype khususnya VoIP ini serta dilakukan evaluasi hasil penelitiannya, maka dapat diambil kesimpulan bahwa telah dibuat aplikasi yang dapat mengidentifikasi lalu-lintas data Skype khususnya VoIP pada sebuah jaringan LAN, dimana didapatkan hasil bahwa aplikasi ini dapat mengetahui *user* mana yang telah melakukan komunikasi VoIP menggunakan Skype dalam sebuah jaringan LAN.

5.2 Saran

Bagi pembaca yang ingin mengembangkan aplikasi identifikasi lalu-lintas data VoIP Skype ini dapat memperbaiki beberapa kekurangan sistem ini, yaitu:

1. Penyempurnaan proses *sniffing* pada saat *wireless connection* karena beberapa kali proses *sniffing* yang menggunakan *wireless* ternyata hasilnya tidak valid.
2. *Scanning* jaringan pada *monitoring* untuk bisa dipercepat prosesnya.
3. Aplikasi identifikasi lalu-lintas data VoIP Skype ini dapat dikembangkan dengan dilakukan uji coba tentang fitur-fitur lain misalnya paket data pada saat *chatting*, *idle*, dan lain-lain.
4. Pelaporan diagram tidak hanya pada paket dan pengguna saja melainkan sampai pada *bandwith* yang digunakan..

DAFTAR PUSTAKA

Arifin, Zaenal. 2005. *Langkah Mudah Membangun Jaringan Komputer*. Yogyakarta: Andi

Baset, Salman A. & Henning Schulzrinne. 2004. *An Analysis of Skype Peer-to-Peer Internet Telephony Protocol.* New York: Department of Computer Science Columbia University

Brance, Philip dkk. 2009. *Rapid Identification of Skype Traffic Flow.* Melbourne: University of Swinburne

Dewo, E. S. 2003. *Bandwidth dan Throughput*. Artikel Populer Ilmu Komputer.com: 1-3.

Ed Title. 2004. *Networking dengan Windows Server 2003*. Yogyakarta: Andi

Feibel, W. 1996. *Encyclopedia Of Networking (Second Edition)*. USA: The Network Press.

Irawan, Jusak & Sukmaaji, Anjik. 2003. *Manajemen Jaringan Komputer (Edisi Pertama)*. Surabaya: Stikom

Knowledge, Raf. 2010. *Trik Memonitor Jaringan*. Jakarta: PT. Elex Media Komputindo

Kurniawan, Wiharsono. 2007. *Jaringan Komputer*. Yogyakarta: Andi Offset

Pangera, Abas Ali. 2008. *Menjadi Administrator Jaringan Nirkabel*. Yogyakarta: Andi

Purbo, O. W. & Taufan, R. 2001. *Manajemen Jaringan TCP/IP*. Jakarta: PT. Elex Media Komputindo.

Riantory, Ragowo. 2008. *Pembuatan Aplikasi Sniffer Untuk Unjuk Kerja Pada Sebuah Jaringan*. Surabaya: Stikom

Sholiq. 2006. *Pemodelan Sistem informasi Berorientasi Objek dengan UML*. Yogyakarta: Graha Ilmu.

Suryadi. 1997. *TCP/IP dan Internet Sebagai Jaringan Komunikasi Global*. Jakarta: PT. Elex Media Komputindo.

Zhang, Dongyan dkk. 2010. *Identification and Analysis of Skype Peer-to-Peer Taffic*. China: Department of Computer of Science & Technology Tsinghua University.